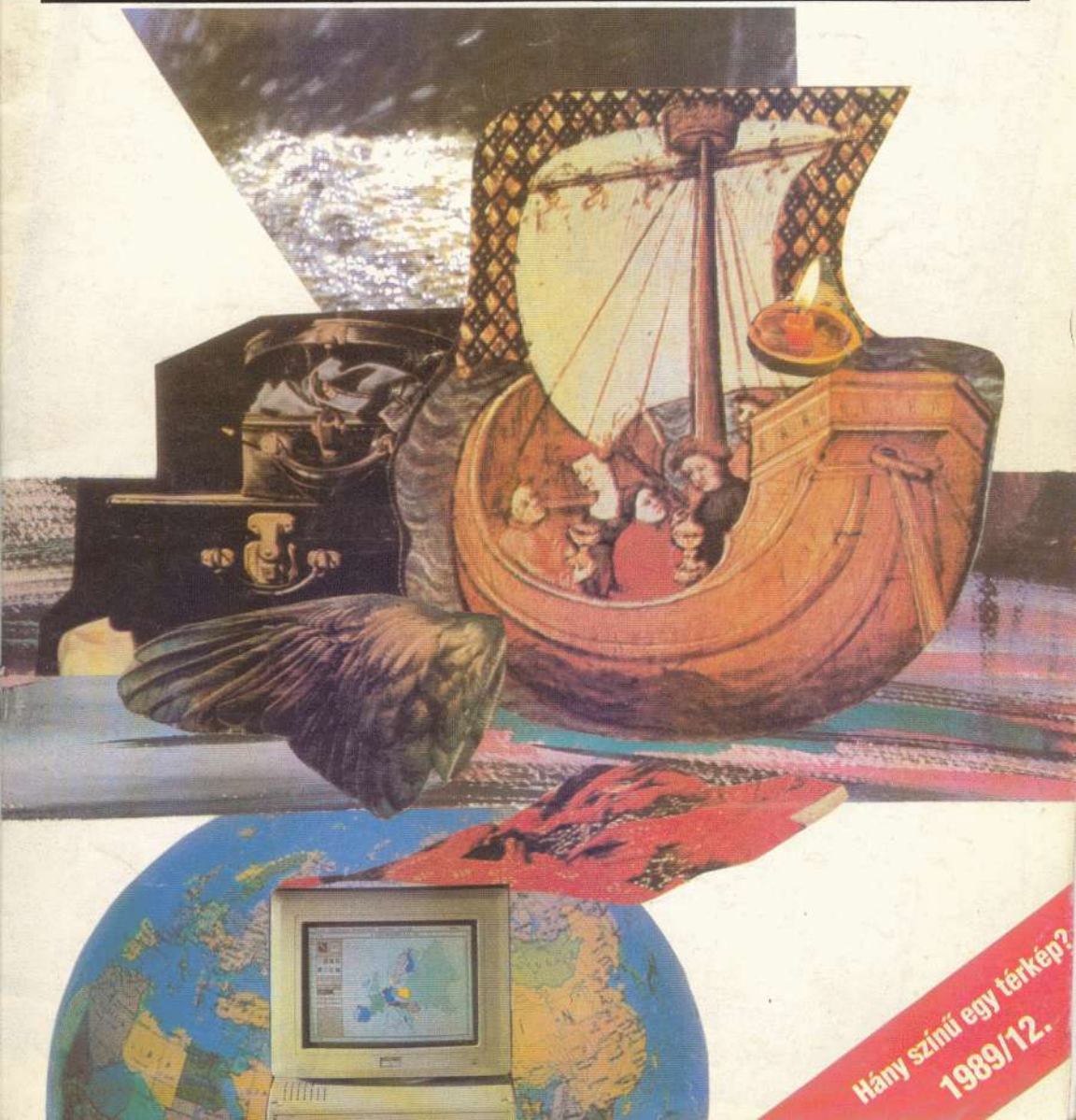
Ára: 30 Ft

μ mikro számítógép magazin



Hány színű egy térkép?

1989/12.

Sci-L

IBM/AT-, 386-KOMPATIBILIS PROFESSZIONÁLIS SZEMÉLYI SZÁMÍTÓGÉPEK

AJÁNLATUNKBAN

ÚJ

P-16/CA-286 AT alapkonfiguráció

80286; 6/10 MHz
640 kbájt RAM
32 Mbájt (ST238)+csatoló
HERCULES csatolókártya
Monokróm monitor, 12"
1,2 Mbájt floppy
Billentyűzet
Soros-párhuzamos interfész
Ár: 150 000,— Ft + ÁFA

P-16/CA-286 AT alapkonfiguráció

80286; 6/10 MHz
1 Mbájt RAM
32 Mbájt (ST238)+csatoló
HERCULES csatolókártya
Monokróm monitor, 12"
1,2 Mbájt floppy
Billentyűzet
Soros-párhuzamos interfész
Ár: 158 000,— Ft + ÁFA

KONFIGURÁCIÓNKÉNT 1 db OPEN ACCESS PROGRAMCSOMAGOT BIZTOSÍTUNK!

P-16/CA-286 AT ALAPKONFIGURÁCIÓ

80286; 10/20 MHz
2 Mbájt RAM
HERCULES csatolókártya
Monokróm monitor, 12"
Billentyűzet (101 gombos)
CDC 94205-51 winchester (40 Mbájt, 28 ms)
+ csatoló
Soros-párhuzamos interfész
Ár: 220 000,— Ft + ÁFA

P-132-386 ALAPKONFIGURÁCIÓ

80386; 20/25 MHz
4 Mbájt RAM
80 Mbájt winchester
Monokróm monitor, 12"
Billentyűzet (101 gombos)
Torony kivitel
HERCULES kártya
Soros-párhuzamos interfész
Ár: 349 000,— Ft + ÁFA

Garancia nélküli reklámáraink:

TRS aszinkron terminál + kábel	9 000,— Ft
TRS mátrixnyomtató + kábel (132 kar/sor)	14 500,— Ft
RANK XEROX 4045 lézernyomtató	290 000,— Ft
UNIBOARD billentyűzet (101 gombos)	4 900,— Ft
TAXAN KPL-710 A/3-as plotter	99 000,— Ft
SECONIC SPL-450 A/3-as plotter	99 000,— Ft
A/3 digitalizáló tablet	110 000,— Ft
ARCHIVE FASTTAPE 60 Mbájt belső streamer + csatoló	69 000,— Ft
60 Mbájt streamer-kazetta	1 000,— Ft
OLIVETTI DY-450 szalagmechanika	1 000,— Ft
EPSON MX-100 festékkazetta	650,— Ft
ACER LP-75 festékkazetta	1 900,— Ft
RIBBON N-15 festékkazetta	1 500,— Ft
	+ÁFA

BŐVÍTÉSI LEHETŐSÉGEINKRŐL KÉRJÉK ÁRJEGYZÉKÜNKET!

SZÁMÍTÁSTECHNIKAI INFORMATIKAI FEJLESZTŐ LEÁNYVÁLLALAT

1011 Budapest, Iskola u. 10.
Telefon: 1-154-065
1-350-180/180, 181, 182, 184

Telefax: 35 39 15
Telex: 22 4599

mikro számítógép magazin

7. ÉVFOLYAM
1989/12. SZÁM

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Bakos Tamás
(programozástechnika)
Broczkó Péter
(hírek)
Kis János
Kovács Győző
(levelezés)
Nagy Imre
(tanuljuk együtt)
Petróczy Judit
(könyvek)
Pinke György
(Enterprise)
Soltészné Vizi Zsuzsa
(tervezőszerkesztő)
Szebenszki Sándor
Tamásné Lakó Erika
Terebessy Ákosné
Varga János
(olvasószerkesztő)

Címképünk:
Velekey József Lajos munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 115-4250

Levélcím:
1371 Budapest Pf. 433

Kiadja:
MTESZ Neumann János Számítógéptudományi Társaság
1054 Budapest, Báthori u. 16.

Levélcím:
1368 Budapest 5. Pf. 240

Telefon: 132-9349

Felelős kiadó:
Tóth Istvánné ügyvezető főtitkárhelyettes

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlap-előfizetési és Lapellátási Irodáján
(1900 Budapest XIII., Lehel u. 10/A)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149
és a Magyar Média
1932 Budapest, Pf. 279
88—1135

89-2735 - Révai Nyomda, Budapest
Felelős vezető:
Horváth Józsefné dr.

INDEX: 25 629
ISSN 0236-6088

TARTALOM

2 Megnyílt a kiállítás!
9 DOS-trükkök
11 Feladatok — megoldások
20 Adom a magyarázatot!
34 A versenyképesség építőkövei
37 Az Amiga programozása assembly nyelven I.
42 A Solarsoft kínálatából
44 Programtermék — Ismét egy tesztgeneráló program
45 Adok — veszek — cserélek

TANULJUK EGYÜTT! 3

3 A Pascal rejtelmei
6 Nem vész el, csak átalakul

CSIPEGETŐ 13

13 ELITE
14 Meghalt C Plus/4?
14 Cracker copy — jó célra
15 Tökéletes karakterkészlet C Plus/4-re
15 The Final Cartridge III.
16 Piaci pillanatfelvétel II.
16 TOP-lista

PROGRAMOZÁSTECHNIKA 17

17 BASIC-bővítések Commodore 16-ra
18 Hány színű egy térkép?
21 Programozási fogások és melléfogások

ENTERPRISE 22

22 Válaszoltak a tulajdonosok
23 Videomonitorként Junoszty
24 Az Enterprise nyolc királynője
25 Mi a manó?

PÉCÉZZÜNK! 26

26 PATHMINDER
31 PC-vizsgáló áramkör
32 Egy másik kulcs…
33 SCRBATCH: képernyőszerkesztő és batch programozást segítő program

SAKK 43

43 Mobilitási függvények

KÖNYVEK — HÍREK — ÉRDEKESSÉGEK 46

AZ OLVASÓ ÍRJA 48

A végül is tető alá került kiállítás a „30 éves a magyar számítógép" elnevezést viseli. Valószínűleg véletlenül maradt el a kiállítás nevéből két fontos szó, hogy „első" és hogy „elektronikus", tudniillik az első magyar, számítógépnek nevezhető eszközt *Kozma László* akadémikus, műegyetemi tanár tervezte és építette 1955-ben, sokkal előbb, mint mi az M—3-at. Sajnos, ez a gép elsősorban oktatási célokat szolgált, és nagyon kis hatással volt a tudományos életre, és méginkább sajnos, hogy gyakorlati alkalmazására Kozma professzor erőfeszítései ellenére szinte egyáltalán nem került sor.

Az első, valóban sokak által, sokféle feladat megoldására használt és elektronikus alkatrészekből épült számítógép az 1959. január 21-én átadott M—3 volt, ezért javasoltam az NJSZT elnökségének, valamint az Országos Műszaki Múzeumnak, hogy ezt a napot válasszuk a magyar számítástechnika „születésnapjának", és ünnepeljük meg egy jubileumi kiállítás megrendezésével.

„A jövőnk helyes megítéléséhez nincs más út, csak az előző fejlődési folyamatok tanulmányozása és a múlt feltárt irányzatainak a kiterjesztése a jelenre. A helyesen értékelt történelem a legjobb felkészülés a jövőre, és egy gondosan létrehozott múzeum meglátogatása az előfeltétele a jövőbe tekintő szellemi kirándulásnak, amely elengedhetetlenül fontos minden sikeres tervezéshez. Ezért igen fontos tényező a számítógép-múzeum, amely híd az informatika korába vezető úton. A számítástechnika gyors változásai feltárják azokat a konstansokat, amelyek a haladás sínjei."

(Részlet Heinz Zemanek professzornak a nyíregyházi kiállítás katalógusába írt előszavából)

Megnyílt a kiállítás!

Nem először írok szerkesztőségi cikket a számítástechnika-történetről, a számítógép-múzeumról, de írtam már — nem is olyan régen — jubileumi kiállításról is. Most azért — még utoljára — megosztanám a legutóbbi élményemet a szeptember 15-i jubileumi kiállítás megnyitásának örömteli pillanatairól.

Nagy helyi és szakmai összefogásból jött létre ez a nagyon szép számítógép-történeti bemutató, amelyre sikerült *Heinz Zemanek* professzort, a nagyon sokak által jól ismert tudóst, a világ első, teljesen félvezetőkből összeállított számítógépének, a MAILÜFTERL-nek a tervezőjét és építőjét is meghívnunk. A kiállítás jellemzésére el kell hogy mondjam Zemanek professzornak egy tréfásnak tűnő, de végül is nagyon komolyan gondolt megjegyzését. Szerinte ugyanis ebben az évben három nevezetes számítógép-történeti kiállítás nyílt vagy fog nyílni a világban. Az egyik a müncheni Deutsches Museumban, a másik itt nálunk, Nyíregyházán, a harmadik pedig az év vége felé Washingtonban, a Smithonian Museumban. Azt hiszem, Nyíregyháza végül is nem került túl rossz társaságba.

Érdekes és már csak nálunk található tárgyakat mutatunk be, mint az M—3 meglévő darabjait, Kozma professzor gépének, a MESZ 1-nek szenzációsan megtervezett vezérlőpultját és az egyik szekrényét, egy valószínűleg már sehol sem látható összeállítást az URAL—2-szekrényekből, RAZDAN-elemeket, egyes darabokat az „újabb" gépekből, mint az ELLIOT 803-asból, a KFKI ICT 1905-ösből, a TPA 1001-es gépből, a SZTAKI GD—71-es grafikus megjelenítőjét, az EMG 836-os számítógépet, de még az SZKI—VIDEOTON kooperációban készült első R10-esből is darabokat, és persze még sok minden mást. Az az igazság, hogy mindent fel sem tudnék sorolni, hiszen a katalógus szerint összesen 62 becses emléktárgyat mutatunk be a kiállítás látogatóinak.

Ennek a kiállításnak lelke van, amelyet lehet, hogy csak én érzek, hiszen majdnem minden kiállítási tárgyhoz emlékem fűződik, és persze azt remélem, hogy ez az érzelmi kötődés majd a látogatókban is kialakul, ha szép lassan, komótosan végigsétálnak a termen, és jó alaposan megnézik ezeket az öreg berendezéseket. Beszélgettem a megnyitásra eljött néhány, középiskolásnak látszó diákkal, akik nem akartak hinni a szemüknek, amikor elmondtuk nekik, hogy az ötvenes és a hatvanas években is ilyen „óriási" teljesítményű gépekkel voltak felszerelve a hazai komoly kutatóintézetek, és ezekkel a gépekkel kellett népgazdasági szintű tervezési vagy éppen bonyolult műszaki feladatokat megoldani. Nehezen viseltem volna el, ha ezek a gyerekek kinevetnek, ezért azt már el sem mertem mondani, hogy az Erzsébet-híd méretezését annak idején az M—3-as gépen ellenőrizték, amelyet ma még az általános iskolás gyerekek sem fogadnának el, ha véletlenül valaki meg akarná ajándékozni őket egy ilyen teljesítményű számítógéppel.

A kiállítás megnyitója után Dömölki Bálint elnökletével egy miniszimpóziumot is szerveztünk, ahol Zemanek professzor, dr. Muszka Dániel és jómagam tartottunk előadást a számítástechnikai múlt egy-egy, általunk átélt fejezetéről. Talán ez a délutáni összejövetel volt az egyetlen disszonáns akkordja ennek a valóban emlékezetes napnak, és az egyetlen nem jól szervezett eseménye, mert talán húszan jöttek el meghallgatni Zemanek professzor előadását. Nyíregyházán — ha jól számolom — legalább három főiskola van, tehát főiskolai hallgatók is vannak, sőt azt is el tudom képzelni, hogy a hallgatók egy részét még a számítástechnika is érdekli. Sajnos úgy látszik, a szervezők ezeknek a hallgatóknak az érdeklődését nem tudták felkelteni, hogy egy ilyen páratlan lehetőséget, mint egy délutáni beszélgetést Heinz Zemanekkel kihasználjanak. Nem tartom nagyon valószínűnek, hogy hasonló találkozóra lesz még lehetőség.

Nagyon örültem, hogy a Compexpo, nevezetesen *Ihrig Péter* igazgató a lehetetlenül rövid határidő ellenére a kiállítás megnyitására megjelentette a képes katalógust, ami nagyon hasznos kiadvány, de nem pótolhatja a szakszerű „tárlatvezetést".

Végül még egy fontos dologról kell szólni, az informatikai vagy ha úgy tetszik, számítástechnika-történeti múzeumról. Valami olyasmire gondolok, mint a müncheni Deutsches Múzeum, de még jobban összeszedve a dolgokat, ami nem lesz nehéz, mert — ezt már említettem — a gyűjteményünk nagyon gazdag és különleges. A mai modern múzeumok ma már nemcsak régiségeket állítanak ki, de bemutatják a legkorszerűbb gépeket is, sőt a látogatóknak megengedik, hogy ezeket a gépeket kipróbálják, dolgozzanak rajtuk, és így ismerjék meg az esetleges jövőbeni munkaeszközük működését. Egy olyan múzeumról álmodom, amelyhez a történelmi berendezéseket bemutató kiállítótermek mellett tágas mikroszámítógép-terem, sok-sok terminál, egy szép szoftver- és számítógépes tananyag- (courseware) könyvtár, klubhelyiség, de számítógéppel felszerelt mintaóvoda is tartozik.

A nyíregyházi, szabolcsi vezetők nagyon szeretnék, ha a kiállítás befejezése után (1991 őszén tervezzük bezárni) már meg lehetne nyitni a végleges múzeumot, ahol részben a most bemutatott, részben pedig a ma még a raktárban tárolt anyagokat lehetne kiállítani. Ha valami miatt nem sikerülne Nyíregyházán megfelelő épületet találnunk, akkor sajnos valahol másutt kell elhelyezést keresnünk.

A kiállítás ebben az évben főleg szombaton és vasárnap lesz nyitva, az igazi „szezon" majd tavasszal kezdődik, hiszen a Sóstói Parkot, ahol a kiállítóterem is van, akkor látogatják igazán a nyíregyháziak és természetesen a környékbeliek is. Azt tervezzük, hogy a szomszédos volt KISZ-iskolában, több turnusban számítástechnika-történeti tábort szervezünk középiskolásoknak, főiskolásoknak és egyetemistáknak, akik részben előadásokat hallgathatnak a korai számítástechnikáról, részben, ha sikerül megszervezni, a gyakorlatban is megismerhetik, hogyan is ment az ötvenes évek számítástechnikája.

Itt hívnám fel az ország iskoláinak is a figyelmét erre a páratlan lehetőségre, arra kérve a számítástechnikát szerető tanárokat, hogy az iskolai kirándulásokat szervezzék Szabolcsba, ahol a diákokkal együtt nemcsak a hazai számítástechnika elmúlt harminc évének történetét tanulmányozhatják, de a környéken páratlan szépségű templomokat is találhatnak, nem beszélve a nyíregyházi skanzenről vagy a felső Tisza csodálatos vidékéről, ami egészen biztosan megér akár egy többnapos kirándulást is. Ha ilyen szándékuk van, azt tanácsolom, hogy a Városi Tanács művelődési osztályát hívják fel, vagy a megyei NJSZT-szervezetet; nem hiszem, hogy ne állnának készséggel az iskolák rendelkezésére.

Végigolvasva ezt az írást, végül szeretnék köszönetet mondani a Megyei Tanács, a megyei NJSZT és más szervezetek vezetőinek és munkatársainak azért a segítségért, amit nyújtottak. Utoljára hagytam a megemlékezést a kiállítás szponzorairól, akik a kiszállításhoz, az installáláshoz, a feliratok elkészítéséhez és minden egyéb kiadásunkhoz a szükséges anyagiakat biztosították. Nagyon sok „kérő" levelet küldtem ki, de csak nagyon kevesen válaszoltak, még kevesebben adtak. Akik adtak, azoknak az intézményeknek a neve kerüljön a „virtuális dicsőségtáblára", és fogadják köszönetemet:

Comporgan, Eszterlánc Rt., Micro (μ) Systems, Művelődési Minisztérium, NJSZT, SZÁMALK, SZTAKI, VIDEOTON, X-BYTE Kisszövetkezet.

A lehetőség, hogy bármelyik intézmény neve a „táblára" felkerüljön, még mindenki részére fennáll. További pénzbeli és más adományokat a kiállítás gazdagítására ezután is szívesen elfogadunk, az adományozók értesítését az NJSZT titkársága örömmel várja. A pénzbeli küldeményeket közérdekű felajánlásként az NJSZT 232—90171 2494 MNB számlaszámra kérjük átutalni.

Végül a kiállítást valamennyi kedves olvasónk figyelmébe is ajánlom: egy kellemes nyíregyházi kirándulást és egyben érdekes kalandozást az elmúlt harminc év számítástechnikai emlékei között. Kérem, érezzék jól magukat, ezt kívánja kedves mindnyájuknak:

Kovács Győző

tanuljuk együtt!

A PASCAL REJTELMEI

16. Hang

A Turbo Pascalban összesen két eljárás van, amely a hangképzést befolyásolja. Az **f** frekvenciájú hang bekapcsolására a

sound (f),

kikapcsolására a

nosound

eljárás szolgál. Az **f**-nek integer kifejezésnek vagy konstansnak kell lennie, értéke a frekvenciát Hz-ben adja meg. A hang kitartásának hosszát a **delay** eljárás segítségével állíthatjuk be. Ennek szintaxisa:

delay (t),

ahol a **t** (ez integer konstans vagy kifejezés) a késleltetés hosszát adja ms-ban. Például a

```
begin
  sound 440);
  delay (1000);
  nosound
end;
```

programrészlet egy normál „a" hangot szólaltat meg kb. 1 másodpercre.

Az *50. ábrán* látható programlista különböző skálák megszólaltatására alkalmas. Alaphangként a normál „a" hangot (f=440 Hz) használja. Erre épülve egy egyoktávos, 12 fokú (dodekafon), egy dúr és egy moll skálát állít elő. A program nem a hangképzés problémái miatt lett meglehetősen terjedelmes, hanem mert igyekeztünk olyan példát bemutatni, amely egyrészt még néhány be nem mutatott Pascal-lehetőséget is tartalmaz, másrészt menüje félig-meddig professzionális kivitelű.

A könnyebb megértéshez sorra vesszük a program egyes — részben vagy egészben újdonságnak tekinthető — részeit.

A kezdetnél lévő konstansdeklarációk a menü elemeinek színeit és a menüelemekben lévő szövegeket tartalmazzák. A menü (lásd a menu eljárást) 5 elemből áll. Ebből 4 passzív, ezek a képernyőn fehér alapon kék betűkkel jelennek meg. Az aktív menüelem kék alapon fehér karakterekből áll. A fehér háttér színkonstansa 7, a kék karakterekké 1, az aktív menüelemnél a színkonstansok: 1, illetve 7. A deklarációban egy menüállapothoz összesen 10 szín tartozik, ezeket tartalmazza a konstansdeklaráció egy sora. Mivel összesen 5 menüállapot van (az 5 menüelem kijelölésének megfelelően), ilyen sorból a deklarációban 5 szerepel.

A **hangok** eljárás az oktáv 13 hangját számítja ki. A zenében járatlanok számára ennek elvét is röviden elmondjuk. Az oktáv hangközfelosztását (440 Hz és 880 Hz) úgy kell elvégezni, hogy a frekvenciák hányadosából (ez jelenleg 2) tizenkettedik gyököt vonunk (ennek a műveletnek az eredménye a **hangkoz** konstans); majd az egyes hangok frekvenciáit az előzőből úgy számítjuk ki, hogy a konstanssal megszorozzuk.

A **szol** eljárás a hangok frekvenciáinak kiírását és a hangok megszólaltatását végzi. Mivel a frekvenciaértékek real típusúak, a **sound** eljárásban paraméterként viszont integer adatnak kell szerepelnie, a **trunc** függvénnyel elvégezzük a real-integer konverziót. (A **trunc** a számból elhagyja a tört részt. Real-integer konverzióra egy másik függvény is létezik: a **round,** ez a kerekítés szabályainak megfelelően működik.

A következő eljárások, a frekvencia, a dodekafon, a dúrskála és a mollskála különösebb magyarázatot nem igényelnek, legfeljebb csak annyit, hogy a frekvenciaértékeket képernyőablakba írják ki.

A **cím** eljárás a programfejléc előállítását végzi.

A **menu** eljárás a menü megjelenítésére és a menüelemek közötti választásra szolgál (a menüben válogatni csak lefelé mozgatással lehet; erre bármelyik kurzorvezérlő billentyű és az ESC alkalmas, ezek kódja ugyanis decimális 27). A kiválasztott funkciót Enterrel lehet meghívni (⇒13).

A főprogram lelke a **case of..end** szerkezet. Ez végzi a menüben szereplő lehetőségeknek megfelelő eljárások hívását. A programban itt-ott fellelhető, látszólag semmilyen szereppel nem bíró **textcolor (0)** utasítások a Pascal-kurzor láthatatlanná tételére szolgálnak.

És ami kimaradt...

Bár olvasóinkat kerek egy éven át „gyötörtük" a Pascallal és annak kapcsán a strukturált programozásra szoktatással, nagyon sok dologról nem esett szó. Ezek közül ízelítőül most néhányat röviden összefoglalunk, a többire viszont sajnos már nem kerül sor.

Elsőnek a szövegkezelésről és az ahhoz kapcsolódó dolgokról szólunk. Ezt a területet az eddigiekben egyáltalán nem érintettük. A sztringekkel végzett műveletekhez a Pascal is ad segítséget, négy standard eljárást és négy standard függvényt.

A **delete(sztring,pozíció,darab)** eljárás a sztringből a megadott pozíciótól kezdődően a megadott darabszámú karaktert törli. A sztringbe beszúrni az **insert(sztring1,sztring2,pozíció)** eljárással lehet. Az eredmény: a sztring2-be a megadott karakterpozíció elé beilleszthetődik a sztring1. Az **str(paraméter,sztring)** a paraméterként megadott kifejezés értékét alakítja sztringgé. A paraméter a formázott kivitelnél (write, writeln) ismertetett formájú is lehet. Az eljárás integer és real értékekre egyaránt érvényes. A **val(sztring,változó,hibahely)** a sztringet numerikus adattá konvertálva, a változóba helyezi el. A hibahely azt adja meg, hogy az átalakítás során a sztring hányadik karaktere volt illegális (nem konvertálható). A változó integer vagy real, a hibahely csak integer típusú lehet.

A **copy(sztring,pozíció,darab)** függvény a sztringből a megadott pozíciótól a megadott darabszámú karaktert másolja át az új sztringbe. A **concat(sztring1,sztring2...)** az új sztringben összefűzi a zárójelben felsoroltakat. Ha az új sztring hosszabb, mint 255 karakter, hibajelzés keletkezik, és az összefűzés természetesen érvénytelen. A **length(sztring)** a sztring karaktereinek számával tér vissza. Az eredmény integer típusú. A **pos(sztring1,sztring2)** a sztring1 első előfordulásának helyét adja meg a sztring2-ben. Az eredmény integer típusú. Ha a sztring1 nem része a sztring2-nek, az eredmény 0.

A következőkben néhány standard eljárást sorolunk fel. A **clreol** a kurzorpozíciótól törli a sort. A **delline** törli azt a sort, amelyben a kurzor található; a kurzorpozíció alatti sorok eggyel feljebb kerülnek. Az **insline** egy üres sort szúr be az elé a sor elé, amelyben a kurzor van. A további sorok eggyel lejjebb kerülnek. Ezzel az eljárással egyszerűen megvalósítható a lefelé scroll.

A standard függvények közül a fontosabbakat felsoroljuk, a nevek szinte önmagukért beszélnek:

abs(kifejezés)
sin(kifejezés)
cos(kifejezés)
arctn(kifejezés)
ln(kifejezés)
exp(kifejezés)
frac(kifejezés) (törtrész-függvény)
int(kifejezés)
chr(kifejezés) (a kifejezés csak integer lehet)
random (real véletlenszám, értéke 0 vagy annál nagyobb és 1-nél kisebb)
random (kifejezés) (a kifejezés csak integer lehet, az eredmény integer véletlenszám, értéke 0 vagy annál nagyobb és a kifejezés értékénél kisebb

A nyelvi elemek bemutatását ezzel be is fejeztük. Azok kedvéért, akik a teljességre törekszenek, felsoroljuk azokat a fontosabb dolgokat is, amelyekről nem volt, és a sorozat keretében már nem is lesz szó. Ezek a következők:

— a **set** halmaz, a **record** és **pointer** (mutató) típusok, illetve az ezekhez kapcsolódó műveletek;
— kapcsolat a DOS-szal;
— gépi kódú programok illesztése és hívása;
— az ún. heap memória;
— a hosszú programoknál jól használható overlay (rátöltő) technika;
— a felhasználó által definiálható függvények (ezek készítése gyakorlatilag azonos az eljárások készítésével);
— néhány egyéb standard eljárás és függvény.

Futási és I/O hibajelzések

A fordítás közben felfedezett hibákról a fordító szöveges üzeneteket küld. Ezek a számítástechnikával foglalkozók számára — ha másképpen nem is, de szótárral biztosan — megérthetők. A futási és I/O hibák viszont csak egy kétkarakteres hibakódként jelentkeznek. Ezeket — legalábbis a legfontosabbakat — fel kell sorolni, illetve a felsoroláshoz némi magyarázatot kell fűzni.

Futási hibajelzések:

01 Lebegőpontos túlcsordulás
02 Osztás 0-val
03 Negatív szám az sqrt függvény argumentumában
04 Negatív szám vagy 0 az ln függvény argumentumában
10 sztringhossz-hiba
11 Hibás sztringindex (kívül az 1..255 tartományon)
90 Indextartomány határának túllépése tömbök esetén
91 Kilépés a megadott tartományból a felsorolási, intervallum stb. (ún. skalár) típusoknál
92 A trunc és a round függvény eredménye az integer tartományon kívüli

I/O hibajelzések:

01 A fájl nem létezik
02 A fájl nincs megnyitva olvasásra
03 A fájl nincs megnyitva írásra
04 A fájl nincs megnyitva
20 Meg nem engedett fájlművelet logikai készüléken
91 Keresés a fájlvég-jel (EOF) után
99 „Váratlan" fájlvég (a fájl fizikai vége EOF nélkül, a fájl hibás, olvasási kísérlet az utolsó EOF után stb.)
F0 Írási hiba a lemezen
F1 A könyvtár megtelt
F2 A fájl mérete túl nagy
FF A fájl eltűnt. Kísérlet egy fájl lezárására, amely nincs a könyvtárban (tipikus hiba: futás közbeni lemezcsere)

Végezetül minden olvasónknak sok sikert kívánok:

Nagy Imre

50. ábra

```
program skala;
const c:array[1..5] of
        array[1..10] of integer=
            ((1,7,7,1,7,1,7,1,7,1),
             (7,1,1,7,7,1,7,1,7,1),
             (7,1,7,1,1,7,7,1,7,1),
             (7,1,7,1,7,1,1,7,7,1),
             (7,1,7,1,7,1,7,1,1,7));
      m:array[1..5] of string[20]=
            ('A hangok frekvenciái',
             'Dodekafon skála     ',
             'A-dur skála         ',
             'A-moll skála        ',
             'Kilépés             ');
label ide;
var alaphang,hangkoz:real;
    i,j:integer;
    ch:char;
    hang:array[1..13] of real;
    h:array[1..9] of real;

procedure hangok;
begin
    alaphang:=440.0;
    hangkoz:=exp(ln(2.0)/12);
    hang[1]:=alaphang;
    for i:=1 to 13 do
        begin
            hang[i+1]:=hang[i]*hangkoz
        end
end;

procedure folytat;
begin
    writeln;
    writeln('Folytatás bármely billentyű'
    read(kbd,ch);clrscr
end;

procedure szol;
begin
    write(h[i]);writeln('  Hz');
    sound(trunc(h[i]));delay(500);
    nosound
end;

procedure frekvencia;
begin
    window(44,1,71,18);
    clrscr;writeln(m[1]);writeln;
    for i:=1 to 13 do
        begin
            write(hang[i]);
            writeln('  Hz')
        end;
    folytat;
    textbackground(0);
    clrscr;window(1,1,79,24)
end;

procedure dodekafon;
```

```
begin
    window(44,1,71,18);
    clrscr;writeln(m[j]);writeln;
    for i:=1 to 13 do
        begin
            h[i]:=hang[i];
            szol
        end;
    folytat;
    textbackground(0);
    clrscr;window(1,1,79,24)
end;

procedure durskala;
begin
    h[1]:=hang[1];
    h[2]:=hang[3];
    h[3]:=hang[5];
    h[4]:=hang[6];
    h[5]:=hang[8];
    h[6]:=hang[10];
    h[7]:=hang[12];
    h[8]:=hang[13];
    window(44,1,71,13);
    clrscr;writeln(m[j]);writeln;
    for i:=1 to 8 do
        szol;
    folytat;
    textbackground(0);
    clrscr;window(1,1,79,24)
end;

procedure mollskala;
begin
    h[1]:=hang[1];
    h[2]:=hang[3];
    h[3]:=hang[4];
    h[4]:=hang[6];
    h[5]:=hang[8];
    h[6]:=hang[9];
    h[7]:=hang[11];
    h[8]:=hang[13];
    window(44,1,71,13);
    clrscr;writeln(m[j]);writeln;
    for i:=1 to 8 do
            szol;
    folytat;
    textbackground(0);
    clrscr;window(1,1,79,24)
end;

procedure kilepes;
begin
    textbackground(0);textcolor(14);
    clrscr;writeln('Program vége');
    halt
end;

procedure cim;
begin
    clrscr;
    textbackground(7);textcolor(4);
    writeln('      Hangdemo        ');
    writeln(' Alaphang: normál a ');
    writeln('                      ');
    j:=1
end;

procedure menu;
begin
    gotoxy(1,4):
    repeat
        for i:=0 to 4 do
        begin
            textbackground(c[j,2*i+1]);
            textcolor(c[j,2*i+2]);
            writeln(m[i+1]);
        end;
        read(kbd,ch);
        if ch=#27 then
            begin
                j:=j+1;
                if j=6 then j:=1
            end;
        gotoxy(1,4);
    until ch=#13;
end;


{FOPROGRAM}
begin
    clrscr;textcolor(0);
    cim;
    hangok;
ide:menu;
        case j of
            1:frekvencia;
            2:dodekafon;
            3:durskala;
            4:mollskala;
            5:kilepes
        end;
    textcolor(0);
    goto ide
end.
```

HIBAIGAZÍTÁS

Az 1989/6. számban megjelent, Más billentyűzet XT-AT-n című cikkem programlistájában sajnálatosan benne maradt egy hiba, ami a szürke „+" és „−" billentyűk alkalmazását nehézkessé teszi. Ez elkerülhető, ha három sort kihagyunk a programból. Ezek: az első oszlop utolsó és a második oszlop első két sora.

```
(CMP AL,32
JC KERES
MOV AH,0)
```

Minden felhasználótól elnézést kérek!

Krizsák László

Az ékezetes Easy Script konverziója

Nem vész el, csak átalakul

A számítógépes környezetben felhalmozódott információbázis tetemes részét alkotják a szövegszerkesztő programokkal készített és tárolt szövegek. Viszont éppen e programok népszerűsége sokféleséget is eredményezett. Szerzőnk a sokféleségből származó egyik jellegzetes probléma áthidalását tűzte ki célul és oldotta meg az itt közölt programmal.

A szövegszerkesztő programok használatának — tapasztalatom szerint — az az egyik jellegzetessége, hogy az alkalmazók általában ragaszkodnak kedvenc szerkesztőjükhöz, nem szívesen tanulják meg egy másiknak a használatát. Különösen igaz ez a C64-re készült különböző „magyarított" Easy Script-változatokra, ahol a programok készítői — még az írógépszabványokra is hivatkozva — furfangosnál furfangosabb helyekre dugták el az ékezetes karaktereket. Ha az alkalmazó hozzájut valamilyen idegen formájú szöveges fájlhoz (leíráshoz, dokumentációhoz stb.), és azt nemcsak nyomtatni, hanem módosítani, javítani is kívánja, akkor két lehetőség között választhat. Az egyik, hogy izzadságos munka árán eredeti szövegszerkesztőjével dolgozza fel a szöveget — ez sokszor gyakorlatilag a teljes fájl újraírását jelenti —, a másik, hogy konvertálja saját megszokott szövegszerkesztőjére.

Ez utóbbi megoldást alkalmazni szándékozók segítésére dolgoztam ki és kívánom közkinccsé tenni a mellékelt programot, amely a hozzám eljutott három magyarított Easy Script-változat szövegfájljainak konverziójára alkalmas. (Az itt nem említett negyediket, az Easy Novót a Magyar Easyvel kompatibilisnek találtam.)

A program végrehajtja az ékezetes karakterek és a formátumkezelő utasítások konverzióját. A sikeres konverzió alapfeltétele, hogy az eredeti szöveg az eredeti szövegszerkesztővel nyomtatható legyen! De ez még nem garancia a konvertált szöveg azonnali nyomtathatóságára a Texterre konvertálás esetén, mert abból néhány utasítás a „magyarítás" során kimaradt (olyanok, mint a CH, LF, PT, LP); igaz, hogy ezekre elméletileg nincs is szükség. Ilyen esetekben a szövegbe utólag bele kell javítani. A másik probléma a szimpla és a dupla idézőjelek kérdése. Az ABC Doku nem ismeri a szimpla idézőjelet, helyére ékezetes karakter került. A Magyar Easy dupla idézőjel után ékezetes magánhangzók helyett zagyvaságokat nyomtat. Ezt a Texternél úgy védték ki, hogy a képernyőn még megjelenik ugyan a kettős idézőjel, de nyomtatásra már csak mint szimpla kerül. Az említettek miatt előfordulhat, hogy „körbekonvertálás" esetén a szimpla idézőjelek eltűnnek, és helyüket duplák foglalják el.

Én a Texter MPS 802-es változatát használom MPS

```
100 REM    -- EASY SCRIPT TIPUSU --
110 REM       -- SZOVEGFAJLOK --
120 REM  -- KONVERTALO PROGRAMJA --
130 REM -- (C) 1988.  KERY LASZLO --
140 REM * ELSO BLOKK *
150 DATA160,0,169,163,141,192,199,141
160 DATA193,199,169,39,141,46,199,141
170 DATA84,199,185,8,197,240,6,32,210
180 DATA255,200,208,245,160,0,185,117
190 DATA197,240,6,32,210,255,200,208
200 DATA245,169,1,133,96,32,110,196
210 DATA32,228,255,201,0,240,249,201
220 DATA17,240,10,201,145,240,20,201
230 DATA13,240,30,208,235,230,96,165
240 DATA96,201,7,208,224,169,1,133
250 DATA96,208,218,198,96,165,96,201
260 DATA0,208,210,169,6,133,96,208
270 DATA204,169,147,32,210,255,169
280 DATA18,32,210,255,162,3,160,6,24
290 DATA32,240,255,165,96,32,150,196
300 DATA162,8,160,3,24,32,240,255,169
310 DATA192,133,97,169,198,133,98,32
320 DATA161,196,170,32,207,255,201
330 DATA13,240,8,157,213,198,232,224
340 DATA16,144,241,224,0,208,3,76,0
350 DATA192,169,44,157,213,198,232
360 DATA169,83,157,213,198,232,169
370 DATA44,157,213,198,232,169,82,157
380 DATA213,198,232,142,234,198,169
390 DATA235,133,97,169,198,133,98,32
400 DATA161,196,170,32,207,255,201
410 DATA13,240,8,157,4,199,232,224
420 DATA16,144,241,224,0,240,133,169
430 DATA44,157,4,199,232,169,83,157
440 DATA4,199,232,169,44,157,4,199
450 DATA232,169,87,157,4,199,232,142
460 DATA25,199,162,0,189,213,198,221
470 DATA4,199,208,10,201,44,240,3,232
480 DATA208,241,76,98,192,162,26,160
490 DATA199,169,1,32,189,255,169,1
500 DATA162,8,160,15,32,186,255,32
510 DATA192,255,32,174,196,173,234
520 DATA198,162,213,160,198,32,189
530 DATA255,169,3,162,8,160,3,32,186
540 DATA255,32,192,255,32,174,196,169
550 DATA0,133,48,133,52,169,16,133
560 DATA49,133,53,162,3,32,198,255
570 DATA160,0,32,207,255,145,52,230
580 DATA52,208,2,230,53,36,144,80,241
590 DATA32,204,255,169,3,32,195,255
600 DATA169,1,32,195,255,165,96,201
610 DATA3,144,6,201,5,144,5, 52373
620 REM * MASODIK BLOKK *
630 DATA176,6,76,237,193,76,97,194
640 DATA201,5,208,51,169,66,133,54
650 DATA169,199,133,55,169,47,133,56
660 DATA169,199,133,57,169,137,133
670 DATA84,169,199,133,85,169,163,133
680 DATA86,169,199,133,87,169,85,133
690 DATA88,169,199,133,89,169,111,133
700 DATA90,169,199,133,91,76,215,194
```

```
710 DATA169,47,133,54,169,199,133,55
720 DATA169,66,133,56,169,199,133,57
730 DATA169,85,133,84,169,199,133,85
740 DATA169,111,133,86,169,199,133
750 DATA87,169,137,133,88,169,199,133
760 DATA89,169,163,133,90,169,199,133
770 DATA91,169,34,141,84,199,76,215
780 DATA194,201,1,208,56,169,179,141
790 DATA192,199,169,28,133,54,169,199
800 DATA133,55,169,66,133,56,169,199
810 DATA133,57,169,85,133,84,169,199
820 DATA133,85,169,111,133,86,169,199
830 DATA133,87,169,137,133,88,169,199
840 DATA133,89,169,163,133,90,169,199
850 DATA133,91,76,215,194,169,179,141
860 DATA193,199,169,66,133,54,169,199
870 DATA133,55,169,28,133,56,169,199
880 DATA133,57,169,137,133,84,169,199
890 DATA133,85,169,163,133,86,169,199
900 DATA133,87,169,85,133,88,169,199
910 DATA133,89,169,111,133,90,169,199
920 DATA133,91,76,215,194,201,3,208
930 DATA56,169,179,141,192,199,169
940 DATA28,133,54,169,199,133,55,169
950 DATA47,133,56,169,199,133,57,169
960 DATA85,133,84,169,199,133,85,169
970 DATA111,133,86,169,199,133,87,169
980 DATA85,133,88,169,199,133,89,169
990 DATA111,133,90,169,199,133,91,76
1000 DATA215,194,169,179,141,193,199
1010 DATA169,47,133,54,169,199,133,55
1020 DATA169,28,133,56,169,199,133,57
1030 DATA169,85,133,84,169,199,133,85
1040 DATA169,111,133,86,169,199,133
1050 DATA87,169,85,133,88,169,199,133
1060 DATA89,169,111,133,90,169,199,133
1070 DATA91,169,34,141,46,199,169,255
1080 DATA133,50,169,15,133,51,  100737
1090 REM * HARMADIK BLOKK *
1100 DATA32,28,196,144,3,76,150,195
1110 DATA201,128,240,6,32,54,196,76
1120 DATA223,194,32,28,196,144,3,76
1130 DATA150,195,201,42,240,14,205,192
1140 DATA199,208,221,173,193,199,160
1150 DATA0,145,50,240,212,32,28,196
1160 DATA144,3,76,150,195,141,189,199
1170 DATA32,28,196,144,3,76,150,195
1180 DATA141,190,199,160,5,173,189,199
1190 DATA209,84,240,6,136,16,249,76
1200 DATA85,195,173,190,199,209,86,240
1210 DATA6,136,16,233,76,85,195,32,71
1220 DATA196,32,28,196,144,3,76,150
1230 DATA195,201,13,240,6,32,54,196
1240 DATA76,64,195,76,223,194,160,25
1250 DATA173,189,199,209,84,240,7,136
1260 DATA192,5,208,247,240,17,173,190
1270 DATA199,209,86,240,7,136,192,5
1280 DATA208,230,240,3,32,71,196,32
1290 DATA28,196,144,3,76,150,195,201
1300 DATA13,240,14,201,59,240,10,201
```

```
1310 DATA58,240,9,32,54,196,76,118,195
1320 DATA76,223,194,76,11,195,169,195
1330 DATA133,97,169,199,133,98,32,161
1340 DATA196,133,204,32,228,255,201
1350 DATA0,240,249,165,207,208,252,169
1360 DATA1,133,204,162,26,160,199,169
1370 DATA1,32,189,255,169,1,162,8,160
1380 DATA15,32,186,255,32,192,255,32
1390 DATA174,196,173,25,199,162,4,160
1400 DATA199,32,189,255,169,3,162,8
1410 DATA160,3,32,186,255,32,192,255
1420 DATA32,174,196,169,0,133,50,169
1430 DATA16,133,51,162,3,32,201,255
1440 DATA160,0,177,50,32,210,255,230
1450 DATA50,208,2,230,51,165,50,197
1460 DATA52,208,239,165,51,197,53,208
1470 DATA233,32,204,255,169,3,32,195
1480 DATA255,32,174,196,169,1,32,195
1490 DATA255,76,0,192,230,50,208,2,230
1500 DATA51,165,50,197,52,208,8,165
1510 DATA51,197,53,208,2,56,96,160,0
1520 DATA177,50,24,96,160,18,209,54
1530 DATA240,4,136,16,249,96,177,56
1540 DATA160,0,145,50,96,177,88,141
1550 DATA189,199,177,90,141,190,199
1560 DATA198,50,165,50,201,255,208,2
1570 DATA 149867
1580 REM * NEGYEDIK BLOKK *
1590 DATA198,51,160,0,173,189,199,145
1600 DATA50,230,50,208,2,230,51,173
1610 DATA190,199,145,50,96,165,96,72
1620 DATA162,1,198,96,208,5,169,18,32
1630 DATA210,255,32,136,196,232,224
1640 DATA7,208,239,104,133,96,96,138
1650 DATA72,10,24,105,6,170,160,6,24
1660 DATA32,240,255,104,170,189,250
1670 DATA196,133,97,189,1,197,133,98
1680 DATA160,0,177,97,240,6,32,210,255
1690 DATA200,208,246,96,162,1,32,198
1700 DATA255,32,207,255,201,48,208,3
1710 DATA76,204,255,72,169,13,32,210
1720 DATA255,32,210,255,32,210,255,104
1730 DATA32,210,255,32,207,255,201,13
1740 DATA208,246,32,204,255,169,3,32
1750 DATA195,255,169,1,32,195,255,169
1760 DATA0,133,204,32,228,255,201,0
1770 DATA240,249,165,207,208,252,169
1780 DATA1,133,204,104,104,76,0,192
1790 DATA255,0,22,48,79,107,138,166
1800 DATA0,198,198,198,198,198,198,13
1810 DATA14,147,17,32,32,32,32,32,32
1820 DATA197,65,83,89,32,211,67,82,73
1830 DATA80,84,32,32,84,73,80,85,83
1840 DATA85,32,83,90,79,86,69,71,69
1850 DATA75,13,13,32,32,32,32,32,32
1860 DATA32,32,32,32,75,79,78,86,69
1870 DATA82,84,65,76,79,32,80,82,79
1880 DATA71,82,65,77,74,65,13,13,32
1890 DATA32,32,32,32,40,67,41,32,49
1900 DATA57,56,56,46,77,65,74,85,83
```

```
1910 DATA32,49,50,46,32,203,197,210
1920 DATA217,32,204,65,83,90,76,79,13
1930 DATA0,164,164,164,164,164,164,164
1940 DATA164,164,164,164,164,164,164,164
1950 DATA164,164,164,164,164,164,164,164
1960 DATA164,164,164,164,164,164,164,164
1970 DATA164,164,164,164,164,164,164,164
1980 DATA164,164,164,164,164,17,17,17
1990 DATA17,17,17,17,17,17,17,17,17,17
2000 DATA17,17,163,163,163,163,163,163
2010 DATA163,163,163,163,163,163,163,163
2020 DATA163,163,163,163,163,163,163,163
2030 DATA163,163,163,163,163,163,163,163
2040 DATA163,163,163,163,163,163,163,163
2050 DATA163,163,163,163,163,163,163,32
2060 DATA32,32,32,32,32, 194625
2070 REM * OTODIK BLOKK *
2080 DATA67,82,83,82,32,76,69,45,70
2090 DATA69,76,32,45,45,62,32,86,65
2100 DATA76,65,83,90,84,65,83,13,32
2110 DATA32,32,32,32,32,82,69,84,85
2120 DATA82,78,32,32,32,32,32,32,45
2130 DATA45,62,32,69,82,86,69,78,89
2140 DATA69,83,73,84,69,83,0,77,65,71
2150 DATA89,65,82,32,69,65,83,89,32
2160 DATA45,45,62,32,84,69,88,84,69
2170 DATA82,146,13,13,0,84,69,88,84
2180 DATA69,82,32,32,32,32,32,32,45
2190 DATA45,62,32,77,65,71,89,65,82
2200 DATA32,69,65,83,89,146,13,13,0
2210 DATA77,65,71,89,65,82,32,69,65
2220 DATA83,89,32,45,45,62,32,65,66
2230 DATA67,32,68,79,75,85,146,13,13
2240 DATA0,65,66,67,32,68,79,75,85,32
2250 DATA32,32,32,45,45,62,32,77,65
2260 DATA71,89,65,82,32,69,65,83,89
2270 DATA146,13,13,0,84,69,88,84,69
2280 DATA82,32,32,32,32,32,32,45,45
2290 DATA62,32,65,66,67,32,68,79,75
2300 DATA85,146,13,13,0,65,66,67,32
2310 DATA68,79,75,85,32,32,32,32,45
2320 DATA45,62,32,84,69,88,84,69,82
2330 DATA146,13,13,0,146,193,32,70,79
2340 DATA82,82,65,83,70,73,76,69,32
2350 DATA78,69,86,69,58,32,0,0,0,0,0
2360 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0
2370 DATA0,0,0,0,13,13,32,32,32,193
2380 DATA32,67,69,76,70,73,76,69,32
2390 DATA78,69,86,69,32,46,46,58,32
2400 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2410 DATA0,0,0,0,0,0,0,0,0,73,0,96,105
2420 DATA97,106,98,107,99,108,100,109
2430 DATA101,110,102,111,103,112,104
2440 DATA113,39,176,221,177,178,182
2450 DATA175,164,186,165,222,171,168
2460 DATA183,39,191,181,179,219,34,176
2470 DATA174,177,178,182,161,171,175
2480 DATA164,186,221,168,191,183,170
2490 DATA165,180,181,39,80,78,72,70
2500 DATA76,67,76,82,83,74,80,84,76
```

```
2510 DATA80,70,77,79,82,67,86,72,72
2520 DATA83,76,80,76,83,66,68,84,75
2530 DATA72,77,77,80,85,76,76,78,35
2540 DATA80,65,70,65,78,80,76,82,65
2550 DATA70,84,80,83,77,70,76,76,67
2560 DATA66,74,83,83,80,83,85,79,76
2570 DATA77,86,74,75,70,70,83,76
2580 DATA80,76,90,83,76,74,75,72,77
2590 DATA77,84,75,72,72,83,35,68,75
2600 DATA69,75,82,69,66,74,65,70,84
2610 DATA80,0,0,0,163,163,0,13,13,13
2620 DATA13,32,32,32,193,90,32,65,84
2630 DATA75,79,68,79,76,65,83,32,75
2640 DATA69,83,90,33,13,32,32,32,203
2650 DATA69,82,69,77,32,65,32,67,69
2660 DATA76,76,69,77,69,90,84,32,33
2670 DATA32,0,0,0,0,0,0,0,0,0,0,0,0
2680 DATA 230908
2690 K=12*4096: S=0: D=0
2700 FOR I=K TO K+373: READ B: S=S+B
2710 POKE I,B: NEXT I: K=I
2720 READ C: D=D+1: IF C <> S THEN2860
2730 FOR I=K TO K+360: READ B: S=S+B
2740 POKE I,B: NEXT I: K=I
2750 READ C: D=D+1: IF C <> S THEN2860
2760 FOR I=K TO K+377: READ B: S=S+B
2770 POKE I,B: NEXT I: K=I
2780 READ C: D=D+1: IF C <> S THEN2860
2790 FOR I=K TO K+383: READ B: S=S+B
2800 POKE I,B: NEXT I: K=I
2810 READ C: D=D+1: IF C <> S THEN2860
2820 FOR I=K TO K+550: READ B: S=S+B
2830 POKE I,B: NEXT I: K=I
2840 READ C: D=D+1: IF C <> S THEN2860
2850 SYS 12*4096
2860 PRINT"HIBA A(Z)"D"[]. BLOKKBAN!"
```

1000-es nyomtatóval a következőkben felsorolt több előnyös tulajdonsága miatt. Egy oldalra 81 sor fér el, és az ékezetek külön nyomtatása ellenére is gyorsabb, mint amikor minden ékezetes karakternek külön „nekifut" a nyomtatófej. Az is fontos szempont, hogy NLQ-módban csak a „két pont" ékezet nem NLQ, és az alatta lévő karakterek azok. Ez a tény már nem annyira zavaró, mint amikor az egész ékezetes karakter nem NLQ, és ráadásul még ki is lóg a sorból.

A konverter használatáról — egyszerű kezelhetősége miatt — csak annyit, hogy ha a forrásfájl nevéhez nem írunk semmit, akkor a menühöz jutunk vissza, és ha a célfájl neve azonos a forrásfájl nevével (vagy üres), a forrásnév beviteléhez kerülünk.

A program assembly nyelven készült, de gyakorlati okokból nem assembly listán mellékelem. Ennek a listának a hossza ugyanis több, mint nyolc sűrűn nyomtatott oldal lenne. Közlésre megfelelőbbnek tartom a hexdump listát vagy a program „datásított" változatát. Azért döntöttem ez utóbbi lehetőség mellett, mert a BASIC program bevitele billentyűzetről és kivitele a háttértárra is egyszerűbb. A DATA sorokat blokkokba szerveztem és védőinformációval láttam el, így a bevitel során esetleg elkövetett hibák felfedezése és javítása is egyszerűbb. A BASIC program természetesen RUN-nal indítható, a konverterre való áttérést az 1360-as sorban lévő SYS 12×4096 utasítás végzi el.

Kéry László

DOS-trükkök

Akik olvasták a Mikroszámítógép Magazinban megjelent Vigyázat! című cikket, azok számára szolgálok két újabb kiegészítéssel.

A legtöbb másoló az olvasási hibákat (20, 21, 22, 27, 29) könnyen másolja. A DOS egyik hibája lehetőséget biztosít néhány másolóprogram életének a megnehezítésére. Azok a másolóprogramok, amelyek a szinkronjelek felismeréséhez a szinkronérzékelő vonalat (syncron detect line, $1C00 7. bit) használják, és az eltelt idő méréséről megfeledkeznek, egy olyan szinkronjelsorozat esetén, amely a teljes sávot betölti, végtelen ciklusba kerülnek. Tehát ha az első sávra ilyet teszünk, akkor ott meg is halnak. Olyan másolók is megfeledkeznek erről, mint például az ULTRABYTE! *(1. lista).*

A szinkronjelek felírását a $FDA3 DOS ROM-rutin végzi. Hatását a program segítségével is megfigyelhetjük *(2. lista).*

A DOS felépítése lehetővé teszi olyan hiba szimulálását is (25 write error), amely csak fizikai lemezhiba esetén fordulhatna elő. Ez a hiba nem a lemez olvasásakor, hanem annak írásakor jelentkezik. A DOS minden adatblokk felírása után visszaolvassa és összehasonlítja az adatokat a pufferben lévőkkel. Eltéréskor ezt a hibát kapjuk.

Minek köszönhetjük ezt a lehetőséget? Nézzük meg egy adatblokk felírásának folyamatát. A header azonosítása után a header gapet (9 bájt) átolvassa, csak ezután kapcsol írásra, és újra felírja az adatblokk-szinkronjeleket is. A header gap tartalma tehát érintetlen marad! Olvasáskor viszont a header azonosítása után azonnal szinronjelet keres. Ha tehát a header gapbe egy szinkronsorozatot írunk, akkor olvasáskor (a kiírás ellenőrzésekor) a DOS az eredeti helyett ezt a sorozatot fogja az adatblokk szinkronsorozataként érzékelni. Így ellenőrzéskor természetesen hiba lép fel. Ez a legtöbb másolónak szintén problémát okoz *(3. lista).*

EREDETI

/	Header-szinkron	Header	GAP	Adat-szinkron	Adatok

MÓDOSÍTOTT

/	Header-szinkron	Header	GAP	+Szinkron	GAP	Adat-szinkron	Adatok
				Adat-szinkron	Adatok		

1. lista

```
0000                    .OBJ   M
0001 ;
0002  DK                =      $0600
0003  DV                =      $0630
0004  DS                =      $060B
0005  CK                =      $C000
0006  TRACK             =      1
0007 ;
0008                    *=     CK
0009 ;
0010  S                 LDA    #<DRIVE
0011                    LDY    #>DRIVE
0012                    STA    $FC
0013                    STY    $FD
0014                    LDA    #<DK
0015                    LDY    #>DK
0016                    STA    $FE
0017                    STY    $FF
0018                    LDA    #<DV
0019                    LDY    #>DV
0020                    STA    $02
0021                    STY    $FB
0022 ;
0023  START             LDA    #8
0024                    JSR    $FFB1
0025                    LDA    #$6F
0026                    JSR    $FF93
0027                    LDA    #"M"
0028                    JSR    $FFA8
0029                    LDA    #"-"
0030                    JSR    $FFA8
0031                    LDA    #"W"
0032                    JSR    $FFA8
0033                    LDA    $FE
0034                    JSR    $FFA8
0035                    LDA    $FF
0036                    JSR    $FFA8
0037                    LDA    #$1E
0038                    JSR    $FFA8
0039 ;
0040                    LDY    #0
0041  SEND              LDA    ($FC),Y
0042                    JSR    $FFA8
0043                    INY
0044                    CPY    #$1E
0045                    BCC    SEND
0046                    JSR    $FFAE
0047                    CLC
0048                    LDA    $FC
0049                    ADC    #$1E
0050                    STA    $FC
0051                    BCC    NX1
0052                    INC    $FD
0053                    CLC
0054  NX1               LDA    $FE
0055                    LDX    $FF
0056                    ADC    #$1E
0057                    STA    $FE
0058                    BCC    NX2
0059                    INC    $FF
0060  NX2               CPX    $FB
0061                    BCC    START
0062                    CMP    $02
0063                    BCC    START
0064 ;
0065                    LDA    #8
0066                    JSR    $FFB1
0067                    LDA    #$6F
0068                    JSR    $FF93
0069                    LDA    #"M"
0070                    JSR    $FFA8
0071                    LDA    #"-"
0072                    JSR    $FFA8
0073                    LDA    #"E"
0074                    JSR    $FFA8
0075                    LDA    #<DS
0076                    JSR    $FFA8
0077                    LDA    #>DS
0078                    JSR    $FFA8
0079                    JSR    $FFAE
0080                    JSR    $FFAB
0081                    RTS
0082 ;
0083 ;
0084  DRIVE             =      *
0085 ;
0086                    JSR    $FDA3
0087                    JSR    $FE00
0088                    LDA    #1
0089                    JMP    $F969
0090 ;
0091  DSTART            LDA    #TRACK
0092                    STA    $0C
0093                    LDA    #0
0094                    STA    $0D
0095                    LDA    #$E0
0096                    STA    $03
0097  WAIT              LDA    $03
0098                    BMI    WAIT
0099                    RTS
0100 ;
0101                    .END
```

2. lista

```
10 OPEN 15,8,15
20 OPEN 2,8,2,"#"
30 PRINT#15,"U1 2 0 1 0"
40 INPUT#15,EN$,EM$,T,S
50 PRINT EN$" "EM$T" "S
60 CLOSE 2
70 CLOSE 15
```

3. lista

```
10 OPEN 15,8,15
20 OPEN 2,8,2,"#"
30 PRINT#2,"TESZT"
40 PRINT#15,"U2 2 0 35 10"
50 INPUT#15,EN$,EM$,T,S
60 PRINT EN$" "EM$T" "S
70 CLOSE 2
80 CLOSE 15
```

A tesztelés a *4. listán* látható programmal végezhető.

Mivel minden másoló működése a szinkronjelek érzékelésén alapul, ezért a kísérletező kedvűeknek a szinkronjelek nélküli adattárolást javasoljuk, mert ez még minden ismert másoló ellen hatásosnak bizonyult.

Bakos Imre—Kelemen Róbert

4. lista

```
0000                    .OBJ   M
0001 ;
0002  DK                =      $0600
0003  DV                =      $0700
0004  DS                =      $0629
0005  CK                =      $C000
0006  TRACK             =      35
0007  SECTOR            =      10
0008 ;
0009                    *=     CK
0010 ;
0011  S                 LDA    #<DRIVE
0012                    LDY    #>DRIVE
0013                    STA    $FC
0014                    STY    $FD
0015                    LDA    #<DK
0016                    LDY    #>DK
0017                    STA    $FE
0018                    STY    $FF
0019                    LDA    #<DV
0020                    LDY    #>DV
0021                    STA    $02
0022                    STY    $FB
0023 ;
0024  START             LDA    #8
0025                    JSR    $FFB1
0026                    LDA    #$6F
0027                    JSR    $FF93
0028                    LDA    #"M"
0029                    JSR    $FFA8
0030                    LDA    #"-"
0031                    JSR    $FFA8
0032                    LDA    #"W"
0033                    JSR    $FFA8
0034                    LDA    $FE
0035                    JSR    $FFA8
0036                    LDA    $FF
0037                    JSR    $FFA8
0038                    LDA    #$1E
0039                    JSR    $FFA8
0040 ;
0041                    LDY    #0
0042  SEND              LDA    ($FC),Y
0043                    JSR    $FFA8
0044                    INY
0045                    CPY    #$1E
0046                    BCC    SEND
0047                    JSR    $FFAE
0048                    CLC
0049                    LDA    $FC
0050                    ADC    #$1E
0051                    STA    $FC
0052                    BCC    NX1
0053                    INC    $FD
0054                    CLC
0055  NX1               LDA    $FE
0056                    LDX    $FF
0057                    ADC    #$1E
0058                    STA    $FE
0059                    BCC    NX2
0060                    INC    $FF
0061  NX2               CPX    $FB
0062                    BCC    START
0063                    CMP    $02
0064                    BCC    START
0065 ;
0066                    LDA    #8
0067                    JSR    $FFB1
0068                    LDA    #$6F
0069                    JSR    $FF93
0070                    LDA    #"M"
0071                    JSR    $FFA8
0072                    LDA    #"-"
0073                    JSR    $FFA8
0074                    LDA    #"E"
0075                    JSR    $FFA8
0076                    LDA    #<DS
0077                    JSR    $FFA8
0078                    LDA    #>DS
0079                    JSR    $FFA8
0080                    JSR    $FFAE
0081                    JSR    $FFAB
0082                    RTS
0083 ;
0084 ;
0085  DRIVE             =      *
0086 ;
0087                    JSR    $FE10
0088                    LDA    #$FF
0089                    STA    $1C03
0090                    LDA    $1C0C
0091                    AND    #$1F
0092                    ORA    #$C0
0093                    STA    $1C0C
0094                    LDX    #5
0095                    LDA    #$FF
0096  WRITE1            BVC    WRITE1
0097                    CLV
0098                    STA    $1C01
0099                    DEX
0100                    BNE    WRITE1
0101  WRITE2            BVC    WRITE2
0102                    JSR    $FE00
0103                    LDA    #1
0104                    JMP    $F969
0105 ;
0106  DSTART            LDA    #TRACK
0107                    STA    $0C
0108                    LDA    #SECTOR
0109                    STA    $0D
0110                    LDA    #$E0
0111                    STA    $03
0112  WAIT              LDA    $03
0113                    BMI    WAIT
0114                    RTS
0115 ;
0116                    .END
```

FELADATOK — MEGOLDÁSOK

Sorozatunkat elsősorban középiskolásoknak szánjuk, de reméljük, hogy minden olvasónknak tanulási lehetőséget és szórakozást nyújt.

A feladatok a Nemes Tihamér országos számítástechnikai verseny színvonalának felelnek meg. Minden esetben olyat választunk, amely röviden, gyorsan megoldható, de a megoldáshoz ötletre van szükség. A megoldást mindig a következő számban közöljük.

Mivel változatosságra törekszünk, különböző programozási nyelveket használunk. Az is előfordul, hogy egy feladatra több programnyelven is közlünk megoldást, ezzel is elősegítve az ismeretszerzést.

A szerkesztőség várja az olvasók, a versenyzők leveleit. A legötletesebb program beküldőjét könyvutalvánnyal jutalmazzuk. Ne feledjenek azonban a programhoz leírást is mellékelni!

18. feladat:

Kétváltozós függvény

Írjon programot, amely kirajzolja egy kétváltozós függvény grafikonját a *4. ábrán* láthatóhoz hasonló módon! A függvény tetszőleges, a (0..1,0,..1,0..1) tartományra normalizált kétváltozós függvény lehet.

Megoldás

Ahhoz, hogy háromdimenziós ábrát ki tudjunk rajzolni a kétdimenziós képernyőn, először ki kell tűzni a tengelyeket. Az *1. ábrán* látható a választott koordináta-rendszer, a tengelyek iránya, a z=0 alapsík és néhány pont képe. Ha ezt az ábrázolási módot választjuk, akkor a három dimenzióból két dimenzióba való transzformáció könnyen elvégezhető:

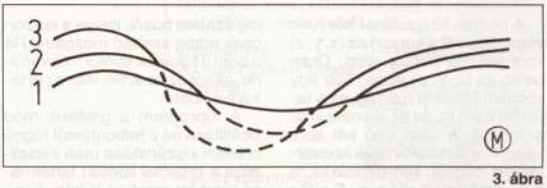
XX=1+x−y
YY=(1−x)+(1−y)+
+ZRel*(1−z)

ahol x, y, z tér-, XX és YY pedig síkbeli koordináták. ZRel — a z irányú nyújtás — értékének helyes megválasztásával a kép valósághűsége még tovább javítható. A közölt programban ZRel=1.

A koordináta-rendszer kitűzése után rátérhetünk a függvény ábrázolására. Egy kétváltozós függvény képe egy görbefelület. Ezt jól láthatóan úgy ábrázolhatjuk, hogy bevonalazzuk két irányból, hasonlóan az 1. ábrán látható alapsík ábrázolásához.

Ha ennyi ismeret alapján megrajzoljuk a függvényt, akkor a Magazin hátsó belső borítóján látható 2. ábrához juthatunk. Bár ez már a függvénynek egy ábrázolása, de mivel a takart vonalak is látszanak, nem túl szemléletes.

Hogyan tudnánk eltüntetni a takart vonalakat? Először vizsgáljuk meg, hogy egy vonal mikor látható! Tegyük fel, hogy a *3. ábrán* látható 1. és 2. görbe van előrébb! Ekkor a 2. görbe szaggatott vonallal rajzolt szakasza nem látható. Hasonlóan, ha a 3. vonal az 1. és a 2. mögött van, akkor annak a szaggatott vonallal rajzolt szakasza takart. A takarás megállapítása tehát igen egyszerű: mindig csak a takarási kontúrt kell nyilvántartani. Ha a görbe ez alá megy, akkor nem látszik.



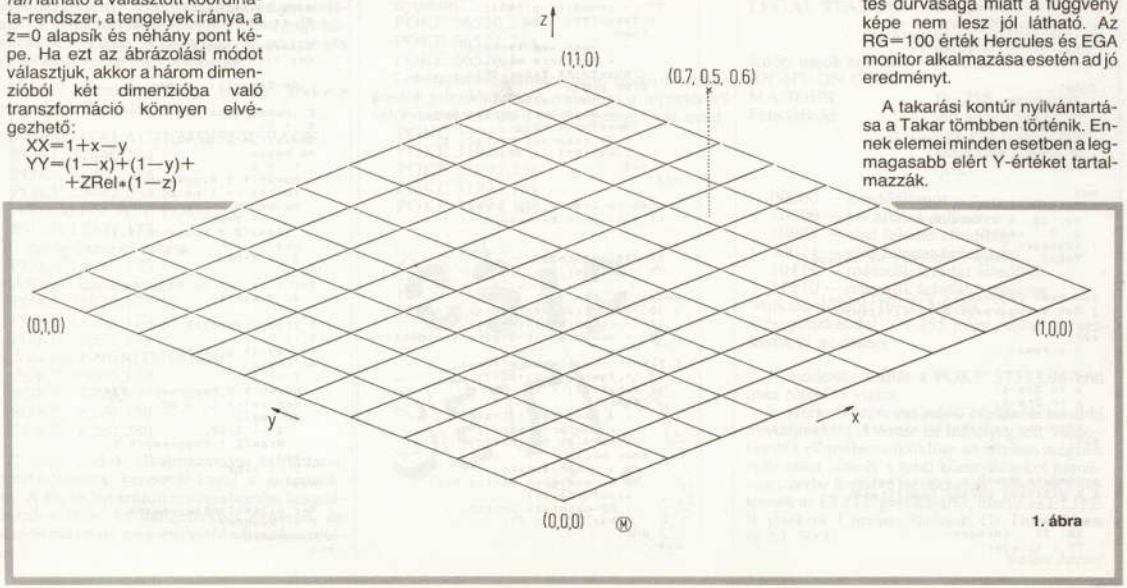
3. ábra

Ily módon juthatunk a *4. ábrán* látható végleges képhez. Ha összehasonlítjuk a 2. és 4. ábrát, rögtön feltűnik ennek az egyszerű takarásvizsgálati módszernek a hibája: nem lehet a sík alá látni, ezért csak olyan függvények ábrázolhatók, amelyek képére rálátunk. Ha az ábrázolt tartományt egy kicsivel nagyobbra választottuk volna, akkor ennél a függvénynél ez a hiba egyáltalán nem jelentkezne. Egy bonyolultabb algoritmussal eldönthető az is, hogy mikor láthatunk a sík alá. Ehhez az alsó takarási kontúrt is tárolni kellene. A közölt program az egyszerű algoritmust használja.

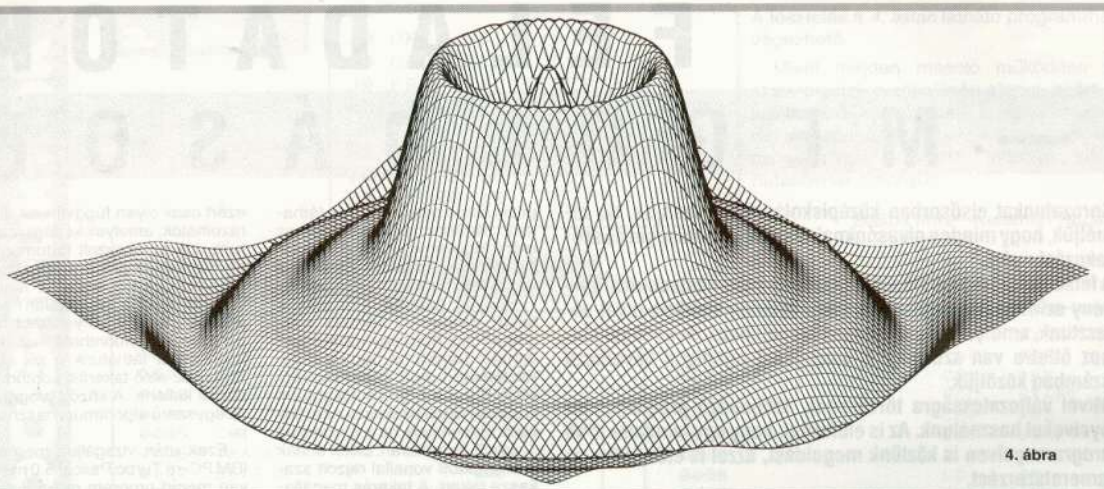
Ezek után vizsgáljuk meg az IBM PC-re Turbo Pascal 5.0 nyelven megírt program működését!

A program elején, a grafikus mód beállításához szükséges változók definiálása után következik RG konstans. Ez határozza meg a görbe ábrázolásához használt vonalak számát. Túl nagy érték választása esetén a rács túl sűrű lesz, a kép zsúfolttá válik, és a futási idő is megnövekszik; túl kis felbontás mellett viszont a közelítés durvasága miatt a függvény képe nem lesz jól látható. Az RG=100 érték Hercules és EGA monitor alkalmazása esetén ad jó eredményt.

A takarási kontúr nyilvántartása a Takar tömbben történik. Ennek elemei minden esetben a legmagasabb elért Y-értéket tartalmazzák.



1. ábra



4. ábra

A pontok kirajzolását két rutin végzi. MoveP a kurzort az (x, y, z) koordinátájú pontra viszi, Draw pedig az (x, y, z) pontig húz egy vonalat. Mindkét rutin figyeli a takarási kontúrt, és az alá nem viszi a kurzort. A rutin első két sora végzi el a térkoordináták konvertálását síkbeli koordinátákká a már ismertetett módon. Ezután annak vizsgálata történik, hogy a pont a már meglevő takarási kontúr alá kerülne-e. Ha a pont ez alá kerülne, akkor a vonalat csak eddig szabad húzni, illetve a kurzort csak eddig szabad mozgatni. Ha a pont a takarási kontúr fölé kerülne, akkor ez az érték lesz az új takarási kontúr.

A főprogram a grafikus mód beállítása és a felbontástól függő értékek kiszámítása után inicializálja a takarási kontúrt tartalmazó tömböt, majd az Y irányú vonalak megrajzolása következik. A függvény értékeit csak a csomópontokban számoljuk, és a közbenső szakaszon egyenes vonalszakaszokkal közelítünk. Ez a számítás időigényét jelentősen csökkenti, és ha jól választottuk meg a felbontást, akkor a képen még nem okoz zavaró torzítást.

Az Y irányú vonalak megrajzolása után következik az X irányú vonalaké. Ez előtt a takarási kontúrt újra be kell állítani.

A program végén a gép billentyűlenyomásra vár, majd a grafikus módból való visszatérés látható.

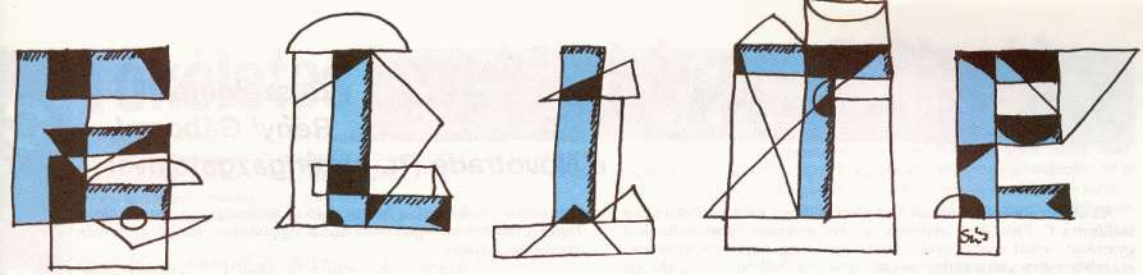
19. feladat:

Keresztreferencia-készítés

Írjon programot, amely beolvas egy assembly nyelvű programlistát, és keresztreferenciát készít róla, azaz kiírja az összes felhasznált címke definiálásának helyét, és felsorol minden címkére való hivatkozást. Ügyeljen arra, hogy a program gyors legyen, és nagy számú címke esetén is működjön, a rendelkezésre álló memóriát hatékonyan használja.

Pintér Gábor

```
program fuggv3D;

uses
  Graph, Crt;

var
{ Grafikus mod: }
  Gd, Gm : integer;
  c      : char;
{ Felbontastol fuggo ertekek: }
  MX, MY   : integer;
  MaxX, MaxY : integer;

const
{ Felbontas: }
  RG=100;
{ Lepeskoz: }
  DX=1/(RG+1); DY=1/(RG+1);
{ Z iranyu rovidules: }
  ZRel=1.0;

var
{ Futo valtozok: }
  ix, iy, i : integer;
  X, Y : real;
{ Eltakart }
  Takar : array [-RG..RG] of integer;


function Fuggveny(x,y : real):real;
{ Ezt a fuggvenyt kell kirajzolni. }

var
  d : real;

begin
  x := 2*x-1;
  y := 2*y-1;
  d := 5*(sqrt(x*x+y*y));
  Fuggveny := (sin(d)-sin(3*d)/3+sin(5*d)/5)/d;
end;


procedure MoveP(x,y,z : real);
{ A grafikus cursor beallitasa. }

var
  XX, YY : integer;
  TT : integer;

begin
{ 3D -> 2D transzformacio: }

  XX := round(MX*(1+x-y));
  YY := round(MY*(2+ZRel-x-y-z*ZRel));

{ Takart vonal vizsgalata }
  TT := round((x-y)*RG);
  if Takar[TT] < YY
    then
      { A pont nem lathato:
        A konturra megy: }
      MoveTo(XX,Takar[TT])
    else begin
      { A pont lathato:
        a szamitott pontra megy: }
      MoveTo(XX,YY);
      { Uj takarasi kontur: }
      Takar[TT] := YY;
    end;
end;

procedure Draw(x,y,z : real);
{ (x,y,z) pont kirajzolasa. }

var
  XX, YY : integer;
  TT : integer;

begin
{ 3D -> 2D transzformacio: }
  XX := round(MX*(1+x-y));
  YY := round(MY*(2+ZRel-x-y-z*ZRel));

{ Takart vonal vizsgalata }
  TT := round((x-y)*RG);
  if Takar[TT] < YY
    then
      { A pont nem lathato:
        A konturig huz: }
      LineTo(XX,Takar[TT])
    else begin
      { A pont lathato:
        a szamitott pontig huz: }
      LineTo(XX,YY);
      { Uj takarasi kontur: }
      Takar[TT] := YY;
    end;
end;

begin
{ Grafikus mod. }
  Gd := Detect;
  InitGraph(Gd, Gm, '');
{ Felbontastol fuggo ertekek: }
  MaxX := GetMaxX;
  MX := trunc(MaxX/2);
  MaxY := GetMaxY;
  MY := trunc(MaxY/(2+ZRel));

{ Takarasi kontur kezdoertek: }
  for i:= -RG to RG
    do Takar[i] := MaxY;

{ Y iranyu vonalak: }
  X := 0;
  for ix := 0 to RG
  do begin
    Y := 0;
    MoveP(X,Y,Fuggveny(X,0));
    for iy := 1 to RG
    do begin
      Y := Y+DY;
      Draw(X,Y,Fuggveny(X,Y));
    end;
    X := X+DX;
  end;
  for i := -RG to RG
    do Takar[i] := MaxY;

{ X iranyu vonalak: }
  Y := 0;
  for iy := 0 to RG
  do begin
    X := 0;
    MoveP(X,Y,Fuggveny(0,Y));
    for ix := 1 to RG
    do begin
      X := X+DX;
      Draw(X,Y,Fuggveny(X,Y));
    end;
    Y := Y+DY;
  end;

{ Var billentyulenyomasra }
  c := ReadKey;
  CloseGraph;
end.
```



Az ELITE című kiváló űrhajós játék valószínűleg sokaknak szerzett örömet és bosszúságot egyaránt. Most néhány tippet adok a játékhoz könnyítésül — vagy éppen nehezítésül — a C64-tulajdonosoknak, és egyben betekintünk a játék háromdimenziós grafikájának rejtelmeibe.

Először a játékot kevésbé ismerőknek adok egy kis tájékoztatást a végrehajtandó feladatokról és a játék kezeléséről. Ez természetesen nem részletes leírás, hiszen inkább a csemegékre, ínyencségekre kívánok kitérni.

A játék lényegében két feladatra tagolódik:

1. el kell érni az ELITE-minősítést,

2. el kell látogatni a 3. galaxis ARREDI nevű bolygójára.

A hosszú utazások során a harcok mellett kereskedéssel növelhetjük vagyonunkat. Az így megszerzett pénzből vásárolhatunk árut, üzemanyagot, fegyvereket és a hajóra kiegészítő berendezéseket, amelyek megkönnyítik a feladatok teljesítését.

Vigyázzunk! A galaxisokban kíméletlen háború folyik a thargoidok és az emberek között az Univerzum feletti uralomért. A thargoidok és emberek ezért kölcsönösen megsemmisítik egymás hajóit, bárhol is találkozzanak. A thargoidok némi előnyre tettek szert a hipertérbeni utazás terén, így egy-egy utazás kellemetlen meglepetéssel végződhet. A thargoidokon kívül állandó veszélyt jelentenek a bolygók körül cirkáló életvidám kalózok. Ezek szintén figyelmeztetés nélkül lőnek mindenre, ami mozog.

A különböző bolygókon eltérő társadalmi berendezkedésű, fejlettségű lakossággal találunk. A velük folytatott kereskedelem nagy körültekintést kíván. Különösen az anarchista bolygókkal veszélyes kereskedni.

Először nézzünk néhány „trainer" kódot a nehézségek leküzdésére:

— Állandó GALACTIC HYPERSPACE:
POKE 31440,234
POKE 31441,234
POKE 23442,234
— Állandó ESCAPE POD:
POKE 12705,173
— Végtelen számú rakéta:
POKE 14010,173
— Állandó üzemanyag:
POKE 31976,173
POKE 12733,173
POKE 8977,173
— Állandó ENERGY BOMB:
POKE 36086,173
POKE 8278,5
POKE 8279,150
POKE 8280,280

A fenti kódok alkalmazásával korlátlanul kereskedhetünk keresztül-kasul a galaxisokban. A korlátlan számú energiabomba használatánál a fény- és hangeffektus elmarad, de minden más hajó megsemmisül a hajó körzetében.

Ezeken a kódokon kívül létrehoztam néhány extrát is:

— Új lézer a bányászlézer erejével és a katonai lézer gyorsaságával:
POKE 45041,234

— Sérthetetlen pajzs:
POKE 34021,173
POKE 34003,173

— Nagy fokozatú hajtómű:
POKE 7976,56
POKE 7977,38
POKE 7978,150
POKE 7979,56
POKE 7980,38
POKE 7981,150
POKE 7982,234
POKE 7983,234
POKE 7989,24
POKE 7990,102
POKE 7991,150
POKE 7992,24
POKE 7993,102
POKE 7994,150

— CORIOLIS bázis felrobbantása lézerrel, energiabombával:
POKE 8283,58
POKE 8541,234
POKE 8542,234

Természetesen vannak olyanok is, akik már túl könnyűnek találják a játékot, és félre is tették. Rájuk is gondoltam néhány olyan POKE-kal, amely nehezíti a feladatok végrehajtását:

— Thargoid-anyahajók bukkannak fel léptennyomon:
POKE 36520,234
POKE 36521,234
POKE 36526,1

— Lehetetlenné válnak a hiperugrások a thargoidok jelenléte miatt, valamint a hipertérből való visszatérés után 6—8 anyahajó vár majd:
POKE 31912,29
POKE 31924,4
POKE 31992,234
POKE 31993,169
POKE 31994,128

Aki ezek után is képes az ELITE-minősítést elérni, az valóban legény az űrhajóján.

Valószínűleg sokaknak megtetszett a játék háromdimenziós grafikája, kitűnő ábrázolásmódja miatt. A forgatások és nagyítások szintén elismerésre méltóak. A 37277-es, illetve a 37304-es címekre 1—33 értékeket írva (a 2 kivételével) feltűnnek a képernyőn a játékban szereplő tárgyak három dimenzióban forgatva. A 37281-es címen pedig a tárgyak nagyítási arányát adhatjuk meg a 0—255 közötti tartományban. A növekedés utolsó, legnagyobb fázisa a 0 értéknél következik be.

Végül szólunk néhány szót a kimentett állás fájlfelépítéséről, változtatásáról és a benne tárolt értékek jelentéséről.

Üzemanyag:	0—255 fényévre
Aktuális galaxis száma:	1—0
Lézerek:	151 Military, 143 Beam, 15 Pulse, 50 Minig, 183 a fent ismertetett „hibrid" lézer
Raktér telítettsége:	0—255 tonna
E. C. M. SYSTEM:	255
FUEL SCOOP:	255
ENERGY BOMB:	127
EXTRA ENERGY UNIT:	1 normál, 2 flotta
DOCKING COMPUTER:	255
GALACTIC HYPERSPACE:	255
LITTLE TRUMBLE JUMPLES:	0—255, 0—255
MISSILE:	0—255 db
LEGAL STATUS:	0-CLEAN 40-OFFENDER 80-FUGITIVE
Kilőtt hajók száma:	0—255
RIGHT ON COMMANDER:	0—255
Feladatkód:	Az alsó 5 bit jelzi egy-egy feladat teljesítését az alábbi módon.

00000 — kezdőállapot
10000 — kis állatok teljesítve
10001 — első feladat elindítva
10011 — első feladat befejezve
10110 — második feladat elindítva
11110 — második feladat befejezve

Minden áruból: 0—255 lehet.
Készpénzkészlet: 0—255 négy bájton helyezkedik el az összes.

A módosított állás a POKE 37373,96 kód után tölthető vissza.

A játékról szívesen adok bárkinek további felvilágosítást. Lemez és felbélyegzett válaszboríték ellenében elküldöm az általam megfúrt változatot, amely a fenti könnyítéseket menürendszerbe foglalva tartalmazza. Szívesen leveleznék az ELITE grafikájáról, illetve az ELITE II. játékról. Címem: Szolnok, Dr. Durst János út 23. 5000.

Bálint József

Meghalt a Plus/4?

Beszélgetés Rényi Gáborral, a Novotrade Rt. vezérigazgatójával

Az utóbbi időben elterjedt az a rémhír, hogy a Commodore cég leállította C Plus/4-es gépének, a „rút kiskacsa" sorozatának a gyártását. Sokak szerint ezután már nem is fogják gyártani a Plus/4-et, tehát a gép „klinikailag halott"-nak minősíthető. Van, aki azt hallotta, hogy a gép hamarosan meg fog jelenni holland exportból, ezúttal *floppyval* együtt. Nos, elérkezett az idő, hogy véget vessünk ennek a bizonytalanságnak, és utánanézzünk, mi is az igazság. Erről kérdeztük Rényi Gábort.

— A Commodore Plus/4 gyártását körülbelül két évvel ezelőtt beszüntette a Commodore cég, de időnként a régi alkatrészekből még összeraktak néhány gépet, ebből volt az utóbbi két évben raktári készlet.

— Ezek után, hogy a gép gyakorlatilag meghalt, várhatunk-e Nyugatról és a Novotrade Rt.-től új szoftvert, hardverkiegészítőket, könyveket?

A programfejlesztés és a könyvkiadás megszűnőben van a Plus/4-hez, úgy értékeljük, hogy ezek a gépek kifutottak. Ezt erősíti meg az utóbbi év tapasztalata is: az általunk fejlesztett számítógépprogramok egyre kevesebb, szinte elenyésző számban kelnek el. A könyvek iránti érdeklődés teljesen visszaesett. Úgy érezzük, hogy a Plus/4-esben rejlő piaci lehetőségeket teljesen kimerítettük. Több száz programot fejlesztettünk, elég sok könyvet csináltunk, különféle kézikönyveket adtunk ki hozzá. Tudomásom szerint Nyugaton sem készülnek ma már hozzá programok.

— Mint tudjuk, az iskolák jelentős részét is ezzel a géppel látták el, éppen a gép előnyös tulajdonságai és olcsósága miatt. Ha ma kellene abban a kérdésben dönteni, hogy milyen gépekkel lássák el az iskolákat, előfordulhatna, hogy más döntés születne?

— Az az igazság, hogy a „szegény ember" — azaz a kevés pénzzel rendelkező hazai iskolarendszer — vízzel főz. Mi kénytelenek voltunk már akkor is abból kiindulni, hogy nagyon kevés a rendelkezésre álló pénz, viszont sok gyereket, nagyon sok iskolát, nagyon sok osztályt kell ellátnunk. Mai eszünkkel, vagy az 1989-es tudásunkkal én úgy gondolom, hogy az iskoláknak Amigát vagy Atari 520 ST-t, esetleg IBM XT-t javasolnánk inkább. Hozzáteszem, hogy az Atari ST és az Amiga eddig COCOM-listán volt, az IBM XT ára pedig viszonylag magas volt az elmúlt években, és ezzel a géppel elsősorban az ipar, a kereskedelem és az államigazgatás igényeinek kielégítése volt a fő cél.

— Mindezeket tudva, továbbra is megoldott-e a Plus/4 és tartozékainak alkatrészellátása és szervize?

— Én úgy tudom, hogy a Plus/4-es alkatrészellátásával nincsenek gondok.

— Térjünk vissza az iskolákhoz! Tudjuk, hogy az iskolák jelentős része ezt a géptípust használja az oktatáshoz, nyilvánvaló, hogy folyamatosan keresik a hozzá való oktatóprogramokat. A Novotrade továbbra is forgalomba hoz programokat iskolák és magánszemélyek részére?

— Az újabb fejlesztések — amibe mi igazán pénzt, eszközt, munkát fektetünk — megszűnőben vannak, tehát valószínű, hogy újabb programok nemigen lesznek. Viszont továbbra is árusítjuk azokat a programokat, amelyeket eddig is forgalmaztunk, ha pedig egy fejlesztőcsoport kidolgoz valamilyen új programot, s szeretné, ha mi forgalmaznánk, ez elől nem fogunk elzárkózni.

— Ön miben látja annak az okát, hogy a C64-es sikeres, a Plus/4-es pedig megbukott a piacon?

— Nem nevezném bukásnak a Plus/4-es esetét. Az a számítógép, amelyből eladtak 300 000 darabot, nem bukott meg. Ha úgy tetszik, a Plus/4-es világméretekben nem tartozik ugyan a legnagyobb sikerek közé, de szeretnék én olyan magyar vállalatot látni, amely 300 000 darab számítógépet el tudott adni a világban. Ráadásul úgy, hogy olyan missziót teljesített vele, miszerint rengeteg gyerek számára biztosította a kezdő számítógépet. Tehát aki ezzel a géptípussal kezdett és megtanul rajta dolgozni, az vágyik majd egy nagyobb teljesítményűre, egy gyorsabbra. Úgy hiszem, hogy az iskolák sem gondolják, hogy számítógépesítésüket befejezték azzal, hogy vettek egy C Plus/4-est. Ennek a gépnek van erkölcsi és fizikai kopása is. Meggyőződésem, hogy 4—5 évig az iskolákban és az oktatásban még alkalmazhatók lesznek. Utána gondolkodni kell, hogy milyen új gép kerüljön be az oktatásba, amely az 1989-es — vagy 1994-es — technológiának felel meg, és nem az 1980-asnak. Mindenki tudja, hogy ebben a szakmában tíz év száz évnek felel meg az ipari fejlődés szempontjából. Tehát ha tíz évvel ezelőtti technikával dolgozunk, azt jelenti, hogy az átlagos elmaradásunk legalább száz év. Azt kell megfontolni, hogy az iskolákba milyen új technikát hozzunk be, amely felülmúlja az IBM PC/XT vagy AT technológiáját, és ugyanakkor árban elfogadható az iskoláknak.

— Ha a gépet esetleg mégis újra gyártanák, vállalná-e a Novotrade az újabb importot?

— A kérdés teljesen illuzórikus, mert a Commodore már régen eldöntötte, hogy ezt a géptípust nem gyártja tovább, nem is foglalkozik vele. Sőt, azon spekulál, hogy a C64 gyártását is be fogja fejezni az elkövetkezendő években, mert ma már az is lefutott technológia. Tehát a Commodore-nál a C64-es és a Plus/4-es helyét az Amiga 500-as fogja elfoglalni. Ma már a nyugat-európai piacon majdnem hasonló árkategóriában lehet kapni egy kvázi 32 bites, sokkal nagyobb felbontóképességű, sokkal gyorsabb, kiváló hanggal rendelkező, a mai követelményeknek megfelelő számítógépet.

B. P.

Cracker copy — jó célra

Szokásunkkal ellentétben ezúttal egy cracker programot közlünk, amely alkalmas ugyan a legtöbb, gyári védelemmel ellátott kazetta másolására, de „békés", legális célokra történő használatával is sokak számára könnyíti meg a munkát. A másoláshoz két gép, két magnó és egy Serial kábel szükséges.

Kössük össze az általunk használt két gépet a Serial kábellel. Ezzel a kábellel köthetjük össze a gépünket egy printerrel vagy floppyval is. Ezt a műveletet egyébként a gépek kikapcsolt állapotában illik végezni. Ezután kapcsoljuk be mindkét gépet. Amelyiknek a magnójában a feltörendő kazetta van, abba az *1. listán* látható programot, amelyikben pedig az üres kazetta van, abba a *2. listán* látható programot gépeljük be. Mindkét programot MONITOR üzemmódban kell begépelni! Ha készen vagyunk, indítsuk el mindkét programot G033C utasítással. Amelyik magnóról másolunk, azon a PLAY gombot, amelyikre másolunk (üres kazetta), azon a PLAY és a RECORD gombot nyomjuk le, mégpedig egyszerre. Ekkor végbemegy a másolás, és kipróbálhatjuk frissen szerzett programunkat.

Bedő Balázs

1. lista

```
. 033C  A5 00     LDA $00
. 033E  09 40     ORA #$40
. 0340  85 00     STA $00
. 0342  78        SEI
. 0343  A9 0B     LDA #$0B
. 0345  8D 06 FF  STA $FF06
. 0348  18        CLC
. 0349  A5 01     LDA $01
. 034B  0A        ASL
. 034C  0A        ASL
. 034D  49 40     EOR #$40
. 034F  85 01     STA $01
. 0351  4C 48 03  JMP $0348
```

2. lista

```
MONITOR

   PC   SR AC XR YR SP
; FF00  00 00 FF 00 F8

. 033C  78        SEI
. 033D  A9 0B     LDA #$0B
. 033F  8D 06 FF  STA $FF06
. 0342  A9 D0     LDA #$D0
. 0344  85 01     STA $01
. 0346  4C 46 03  JMP $0346
```

Tökéletes karakterkészlet C Plus/4-re

Ha a C Plus/4-en új karakterkészletet használunk, azt tapasztaljuk, hogy hiba esetén a képernyőn érdekes, absztrakt karakterek jelennek meg. Ilyenkor csak a RESET vagy az $FF12—FF13 címek újbóli feltöltése jelenthet megoldást.

Ennek oka a következő. Amikor hibás utasításhoz ér a gép, a hibaüzenet kódszámát az akkumulátorba teszi, és elugrik a $8683-as címre. Itt végrehajtódik egy JSR SC7C9 utasítás (VIDEO-RESET). Nézzük meg, hogy mi történik:

```
C7E3 A0 12 FF LDA $FF12:      kiolvasás
C7E6 09 04 ORA+S04:           2. bit magasra
C7EB 80 12 FF STA $FF12:      beírás
RTS
```

Az $FF12 regiszter második bitje azt határozza meg, hogy ROM-ban vagy RAM-ban van a karaktergenerátor helye. A VIDEO-RESET minden egyes hibaüzenetnél ROM-ra állítja. Viszont az $FF03-on lévő regiszter nem lesz alaphelyzetben, pedig ez határozza meg a karakterkészlet helyét. Így a ROM-ban keresi az általunk — mondjuk — $1000-re helyezett karakterkészletet, és mivel ez a terület a RAM, következménye a képernyőn található, modern művészek által is megcsodált összevisszaság. Eddig a magyarázat, de hol a megoldás?

A megoldás a következő. A ROM-ba természetesen nem írhatunk, de a párhuzamosan alatta húzódó RAM-ba igen! Tehát csak egyszerűen a $D000—$D7FF-ig terjedő RAM-ba kell másolni. Ez a látszatnál sokkal egyszerűbben történik. Ha — mondjuk — karakterkészletünk $1000—S17FF-ig terjed, akkor a következőképpen másolhatunk gépi kódban:

```
T 1000 17FF D000
```

Ennek a parancsnak látszólag semmi hatása nincs, hiszen ha a D000-tól kezdődő területet megvizsgáljuk, akkor a gyári karakterkészletet látjuk. Ez annak a következménye, hogy a ROM van felülre lapozva. Annak ellenére, hogy nem látható, az alatta levő RAM-ban megtalálható a mi karakterkészletünk. Erről egyszerűen meggyőződhetünk, ha az $FF12-es regiszter 2. bitjét alacsonyra (0) állítjuk. Ez a következő programmal érhető el gépi kódban:

```
LDA $FF12
EOR +S04
STA $FF12
```

Így már az új karakterkészletet használhatjuk. Mikor „X" paranccsal átlépünk BASIC-be, akkor is végrehajtódik egy VIDEO-RESET, de most már nem telik meg a képernyő különböző értelmezhetetlen karakterekkel, hanem a ROM-karaktereket láthatjuk. Ha újra az általunk tervezett karakterkészletet akarjuk látni, akkor csak a fent említett programot kell használni.

Bessenyei György

THE FINAL CARTRIDGE III.

C64, C64C vagy C-128 típusú gépünket a FINAL CARTRIDGE III. (továbbiakban FC) használatával egy új 64 kbájtos, a memóriát csak kis részben igénybe vevő, sokoldalú rendszerrel bővíthetjük. A GEOS-hoz hasonló menüvezérelt ablaktechnika kényelmessé és gyorssá teszi a sokféle funkció használatát. Az FC-vel a BASIC-bővítő a játéktörésen át a saját- és lemezmemória-trükkökig sok — eddig csak bonyolult rutinok és programok segítségével elvégezhető — feladatot kényelmesen megoldhatunk.

A kártya operációs rendszere alapvetően egy főmenüből (DESKTOP) és a belőle elágazó, illetve egymásból hívható almenükből áll. A DESKTOP INFO két almenüt tartalmaz, amelyben a kártya tervezőinek nevét — DESKTOP almenü —, illetve az aktuális verziószámot — VERSION almenü — tudjuk meg. Ez utóbbi azért lényeges, mert a kártya állandó fejlesztés alatt van. Például a jelenleg kaphatókban még nincs kifejlesztve a DLINK és TLINK almenühöz tartozó szoftver, és így ezek nem is használhatók, de erről majd később.

A SYSTEM almenüből kiléphetünk BASIC-be, ami már természetesen nem a jó öreg V2.0-ás, hanem egy bővített változat. Az új utasítások és parancsok általában azok, mint a többi hasonló bővítésben (például SIMON'S BASIC-ben) eltekintve néhány extrától. Ilyen például a MEM parancs, amely a BASIC memória részletes felosztását mutatja meg, vagy az MREAD-MWRITE utasításpár, amelynek segítségével a ROM alatti 24 kbájt RAM-ot használhatjuk egyszerűen.

A kártya írói gondoltak a monstre programok híveire is, ezért a PACKUNPACK parancspárral elvégezhetjük a programunk tömörítését, illetve kicsomagolását. Az I/O parancsok is változtak némileg, na nem alakjukban, hanem a művelet végrehajtásának sebességében. A kazettás egységgel rendelkezőknek a többi forgalomban levő turbóval kompatibilis, tízszeres töltési, illetve mentési sebességgel rendelkező turbót írtak. Ennek előhívása ugyanúgy a LOAD-SAVE utasításokkal történik, csak egy ,7-et kell mögé biggyeszteni. A lemezes töltés tizenötszörös, a mentés hétszeres sebességű lehet.

Az utasításkészlet kiterjesztése mellett megnövelték a BASIC-szerkesztő hatásfokát is. A kurzorgombokkal lehetőség van a programlista felalá görgetésére, és a CTRL-RETURN billentyűk egyszerre lenyomására az aktuális képernyőtartalomról hardcopy készül.

Az új BASIC-szerkesztőből a MON paranccsal átléphetünk monitor üzemmódba. A FINAL KILL menüpont hivatása egyszerű, ugyanis a kártya kikapcsolására szolgál, aminek megtörténtét a kártyán levő LED kialvása jelzi. Az újraélesztés RESET-tel vagy FREEZE-zel valósítható meg. A FREEZER almenübe a SYSTEM menüből, illetve a FREEZER gomb megnyomásával léphetünk be. Ebben a menüben nyílik lehetőség e betöltött, elindított és lefagyasztott program módosítására, kimentésére és továbbindítására. A BACKUP almenü segítségével menthetjük a programot szalagra, illetve lemezre (TAPE, DISK ablakok). Ha turbózva kívánjuk menteni, akkor az F DISK és F TAPE parancsot kell használnunk.

A GAME almenü nyújt segítséget a játékprogramok egyszerűbbé tételére. A SPRITE I parancscsal sprite-sprite, a SPRITE II-vel pedig a sprite-háttér ütközésének figyelését kapcsolhatjuk ki a tárban levő játékprogramban.

A JOYSWAP és az AUTOFIRE almenük a botkormány kezelésébe segítenek be. JOYSWAP-pel megváltoztathatjuk a játék által használt aktuális portot, az AUTOFIRE pedig gyorstüzeléssel ruházza fel az ezzel nem rendelkező botkormányokat. A COLORS segítségével az aktuális keret-, háttér-, előtérszínek változtathatók tetszés szerint.

A PRINT menü a kártya egyik leghasznosabb funkciójának, a hardcopy-készítőnek a főmenüje. A SETTINGS almenüben kell beállítanunk a nyomtatási paramétereket.

Lehetőség van függőleges, illetve vízszintes nyomtatásra, 1—4-szeres sűrűségű (60—240 dpi) nyomtatásra, 8 vagy 24 tűs nyomtatón, normálba vagy inverzbe, grafikus vagy normál képernyő alapján. A VIEW almenüből megtekinthetjük a nyomtatni kívánt képet. Ezen belül beállíthatjuk a HIRES, illetve MULTICOLOR kép színeit az 53280—53284 regiszterek értékének módosításával.

EXIT-tel visszatérhetünk a FREEZER menübe. A PSET menü aktivizálásával újraéleszthetjük a CENTRONICS printer illesztését a vektorok ismételt beállításával. A RESET és EXIT almenüből a különböző átlépésekre nyílik lehetőség, például a beépített monitorba vagy vissza a DESKTOP-ra. A PROJECT menüben a feltüntetett három almenüből csak a NOTEPAD működik. A NOTEPAD neve ellenére egy egyszerű szövegszerkesztő, néhány érdekes kiegészítéssel. Ilyen például a sortörlés, a vastagbetűs megjelenítés, a sortávolság-állítás. A UTILITES menüben a GEOS PREFERENCE MANAGER-hez hasonló paraméter-beállító menüt és kalkulátort is találunk. A DISK almenüben a szokásos DOS-parancsok mellett megtalálható néhány hasznos tartalomjegyzék-rendező funkció is. Tagoltabbá tehetjük a lemez tartalomjegyzékét a fájlok nevének felcserélésével és elválasztó vonalak beiktatásával. A TAPE almenüben választhatunk a turbós, illetve a normál betöltés között.

A BASIC PREFERENCES-ben néhány; a BASIC-szerkesztő használatát megkönnyítő paramétert állíthatunk be. Például a lenyomott billentyű hangjelzését, az egységszám-beállítást, a kurzorvillogás be-ki kapcsolását és egyéb hasznos dolgokat. A CLOCK menüben az órát állíthatjuk be, mely megjelenik a bal felső sarokban, ha a DESKTOP menüben időzünk. Az óra mellett figyelmeztető ébresztőórát is beállíthatunk a kívánt időpontra. A beépített monitorprogram legtöbb utasítását tekintve hasonlít a magas szintű monitorprogramhoz, de tartalmaz néhány különlegesen hasznos funkciót is. Ilyen például a DISK RAM-ba való írás lehetősége, amikor kiolvashatjuk a lemezegység 16 kbájtos ROM-ját C000-FFFF-ig,illetve írhatunk a 0000-07FF-ig terjedő 2 kbájt RAM-ba. Ezenkívül beavatkozhatunk a két VIA chip buszvezérlőjének munkájába is. A VIA1 az 1B00-1B0F címeken érhető el, a VIA2 pedig az 1C00-1C0F terjedő címeken.

Hasonlóan hasznos funkció a ROM alatti RAM-ok bekapcsolása. Ezzel a módszerrel elérhető a maximális 64 kbájt RAM. A monitor tartalmaz még egy sprite-kereső funkciót is, melynek használatával a sprite-ok megtalálhatók és módosíthatók. Ugyanez megvan a karakterkészletekre is. Tartalmaz még ezenkívül néhány DISK-MONITOR funkciót is. Beolvashatunk egy tetszőleges blokkot a tár tetszőleges címére, változtathatjuk azt és visszaírhatjuk a lemezre.

Összefoglalva: az FC kártya igen hasznos, kellemes segítség. Bár jelenleg nem forgalmazzák, sok ügyes kezű hardveres elkészíti.

Tass Csaba

PIACI

PILLANATFELVÉTEL II.

Folytattuk bolyongásunkat, és újabb két üzletbe látogattunk el. Amint az *1. táblázatból* látható, a Computer—M üzletében gazdag a kínálat lemezekből, festékszalagokból, floppytartókból. Ami nincs benne a táblázatban: kínálnak egy- és többpéldányos leporellót, IBM PC-konfigurációkat, tartozékokat is.

A *2. táblázatban* a Magyar u. 1. szám alatt található Fotoelektronik-Novotrade üzlet kínálata látható. Ehhez az üzlethez igen közel van egy Commodore szerviz is. Elmondhatjuk, hogy itt is kielégítő kínálatot láttunk. Amit itt sem tudunk táblázatba foglalni: kapható Commodore-64, illetve Commodore-128 típusú gép is. **B. P.**

1. táblázat

COMPUTER—M

Termék neve	Ára
EAGLE 1 oldalas, 5"-os floppylemez	80 Ft/db
EAGLE 2 oldalas, 5"-os floppylemez	95 Ft/db
PARROT 1 oldalas, 5"-os floppylemez	110 Ft/db
PARROT 2 oldalas, 5"-os floppylemez	130 Ft/db
PARROT 96 TPI-s, 5"-os floppylemez	180 Ft/db
PARROT HIGH DENSITY, 5"-os floppylemez	320 Ft/db
PARROT 1 oldalas, 3"-os floppylemez	280 Ft/db
PARROT 2 oldalas, 3"-os floppylemez	300 Ft/db
AUTOPROTECT 1 oldalas, 5"-os floppylemez	150 Ft/db
3M 2 oldalas, 5"-os floppylemez (360 k-ra formattálható)	3000 Ft/cs
3M 2 oldalas, 5"-os floppylemez (1,2 Mbájtra formattálható)	5500 Ft/cs
PARROT 1 oldalas, 8"-os, duplasűrűségű floppylemez	220 Ft/db
PARROT 2 oldalas, 8"-os, duplasűrűségű floppylemez	240 Ft/db
Festékszalag MPS 801 nyomtatóhoz	840 Ft/db
Festékszalag MPS 802 nyomtatóhoz	840 Ft/db
Festékszalag MPS 803 nyomtatóhoz	835 Ft/db
Festékszalag MANNESMANN TALLY 490 nyomtatóhoz	1630 Ft/db
Festékszalag MANNESMANN TALLY 86 nyomtatóhoz	1065 Ft/db
Festékszalag MANNESMANN TALLY 300 nyomtatóhoz	3500 Ft/db
Festékszalag CITIZEN 120—D nyomtatóhoz	1200 Ft/db
Festékszalag SEIKOSHA SP 180 nyomtatóhoz	1200 Ft/db
Floppylemez-tartó doboz, 10 db-os, 5"-os	300 Ft/db
Floppylemez-tartó doboz, 60 db-os, 5"-os	1200 Ft/db
Floppylemez-tartó doboz, 100 db-os, 5"-os	1750 Ft/db
MONITOR-szűrőbetét, 12"-os	5000 Ft/db
Játékkazetták és lemezek	300—500 Ft/db

2. táblázat

FOTOELEKTRONIK-NOVOTRADE GT

Termék neve	Ára
CITIZEN 120—D nyomtató	40 000 Ft
Színes monitor Commodore gépekhez	23 000 Ft
IBM AT konfiguráció	200 000 Ft
EPSON LX—42 nyomtató IBM PC-hez	70 000 Ft
3M, 5"-os floppylemez	2 000 Ft/cs
3M, 5"-os floppylemez, HIGH DENSITY	3 500 Ft/cs
JOYSTICK Commodore-64-hez	1 300 Ft
Mikrokapcsolós JOYSTICK Commodore 64-hez	1 600 Ft
JOYSTICK IBM PC-hez	5 000 Ft
Commodore Plus/4 magnóval	10 000 Ft
1551 DRIVE, Plus/4-hez	24 000 Ft
Festékszalag EPSON nyomtatókhoz	1000—1600 Ft/db

TOP LISTA

	játék programok	IBM	AMIGA	C-128	C-64	C+4(16)	SPECTR.	ENTERP.	TVC	APPLE
1.	Pool of Radiance	●	●							
2.	Bard's Tale III.		●		●					●
3.	Neuromancer		●		●					●
4.	Project Firestart				●		●			
5.	Riziko				●					
6.	Zak: Mc Kracken		●		●					
7.	Space Quest II	●								●
8.	Outrun		●		●		●			●
9.	Sentinel	●					●			●
10.	Technocop		●		●		●			●

	felhasználói programok	IBM	AMIGA	C-128	C-64	C+4(16)	SPECTR.	ENTERP.	TVC	APPLE
1.	Black Beard	●								
2.	Geos 2.1				●					
3.	Giga Paint				●					
4.	Amica Paint				●					
5.	Art Studio				●		●			
6.	PC-Write	●								●
7.	Texter				●					
8.	News Room	●	●		●					●
9.	Printmaster	●			●					●
10.	Renegade				●					

Listánkat felhasználói, illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrumra, Enterprise-ra, TVC-re, Atarira és IBM-re készült programrangsorokat várunk havonta.

Címünk:
Mikroszámítógép Magazin
Szerkesztősége
1371 Budapest, Pf. 433
Diákszerkesztőség

BASIC-bővítések Commodore 16-ra

3. rész

Az előző két részben szó volt arról, hogy milyen módszerrel tudjuk gépünk BASIC-tudását megnövelni. Most nézzünk egy konkrét példát!

Az ERASE rutin

A Commodore gépeken, ha egy tömböt már dimenzionáltunk, akkor azt újra nem lehet létrehozni, pedig néha szükség lenne rá. Vessünk egy pillantást az *1. ábrára!* Ez a program 50 nevet olvas be, majd sorrendbe rakja azokat. De mi történik akkor, ha mi 70 nevet akarunk rendezni? Vagy átírjuk a programban a dimenzionáló számot (de ki tudja előre, hogy mennyi név lesz? Arról nem is beszélve, hogy ezeket újra be kell gépelni!), vagy olyan programot készítünk, amely felkészült, és lehetővé teszi a tömb újradimenzionálását.

Az ERASE rutin „eltünteti" a kívánt tömböt, mégpedig oly módon, hogy a törölni kívánt tömb végétől a tömbmutató ($31) által jelzett határig terjedő területet lecsúsztatja a törölni kívánt tömb elejéig, és azt eltünteti fizikailag is. (Nem foglal külön helyet, nem tartja fel a szemétgyűjtő programot.) Ezután a tömböt természetesen újra dimenzionálhatjuk.

Most ismerkedjünk meg az ERASE rutinnal, mint BASIC-bővítéssel *(2. ábra)*.

Az előző részben közölt COMPUTER parancsot most elhagytam, az csak a demonstráció kedvéért készült.

Lássunk egy példát az új parancs alkalmazására, bővítsük ki az alappéldát *(3. ábra)!* A törlő szubrutin azt a célt szolgálja, hogy a beviteli részben használt tömböt a szükséges mértékig „megnyújtsuk". Az elemszám most 10-zel növekszik, de ez bármekkora értékű lehet. Arra kell figyelni, hogy nem lehet negatív. Ha a dimenzionálásra kerülő tömb nagyobb helyet foglal, mint a rendelkezésre álló memória, akkor ?OUT OF MEMORY hibaüzenetet kapunk, bár megfelelő programrész beiktatásával ez kivédhető.

A parancs szintaxisa egyszerű: ERASE A$(X), ahol X akkora lehet, hogy a tömbelemszámnak megfelelő legyen. Erre nagyon alkalmas egy konstans. Például az ERASE A$(1) parancs minden esetben törli a tömböt, és minden olyan helyen alkalmazható, ahol a már dimenzionált tömböt bővíteni szeretnénk. Az ERASE rutin végrehajtó része (nem BASIC-bővítésként) a Magazin 1988/11. számában szerepelt már, itt csak a teljesség kedvéért közöljük.

Kádár Sándor

1. ábra

```
100 dim a$(50)
110 for i=1 to 50
120 input "nev ";a$(i)
130 next:i
140 gosub 180:rem rnedezes
150 gosub 240:rem kiiras
160 end
170 :
180 rem rendezo algoritmus
190 rem
200 rem
210 :
220 return
230 :
240 rem kiiro algoritmus
250 rem
260 rem
270 :
280 return
```

2. ábra

```
ass -16: pass 1 2
 2000                    .opt p
 2000                    x=   $2000
 2000           tok      =    $030c
 2000           detok    =    $030e
 2000           vegre    =    $0310
 2000           adress   =    $96a5
 2000           tvege    =    $31
 2000           charset  =    $0473
 2000           tcim     =    $5f
 2000           kov      =    $47
 2000           tmp1     =    $e1
 2000           be1      =    $0b
 2000           be2      =    $49
 2000           be3      =    $23
 2000 a9 1e             lda  #<tok1
 2002 8d 0c 03          sta  tok
 2005 a9 20             lda  #>tok1
 2007 8d 0d 03          sta  tok+1
 200a a9 32             lda  #<detok1
 200c 8d 0e 03          sta  detok
 200f a9 20             lda  #>detok1
 2011 8d 0f 03          sta  detok+1
 2014 a9 3e             lda  #<vegre1
 2016 8d 10 03          sta  vegre
 2019 a9 20             lda  #>vegre1

 201b 8d 11 03          sta  vegre+1
 201d 60                rts
                        ;
                        ;
 201e 48        tok1    pha
 201f a9 20             lda  #>tabla
 2021 a0 4e             ldy  #<tabla
 2023 20 07 8a          jsr  $8a07
 2026 68                pla
 2027 90 06             bcc  vege
 2029 a5 0b             lda  be1
 202b 48                pha
 202c 4c d6 89          jmp  $89d6
 202f 4c 6c 89 vege     jmp  $896c

 2032 aa        detok1  tax
 2033 84 49             sty  be2
 2035 a0 20             ldy  #>tabla
 2037 84 23             sty  be3
 2039 a0 4e             ldy  #<tabla
 203b 4c 9c 8b          jmp  $8b9c
                        ;
                        ;
 203e 38        vegre1  sec
 203f e9 80             sbc  #$80
 2041 0a                asl  a
 2042 a8                tay
 2043 b9 55 20          lda  innen+1,y
 2046 48                pha
 2047 b9 54 20          lda  innen,y
 204a 48                pha
 204b 4c 73 04          jmp  $0473
                        ;
                        ;
 204e 45 52 41 tabla    .asc "erasE"
 2053 00                .byte 0
                        ;
                        ;
 2054 20 a5 96 innen    jsr  adress
 2057 a0 02             ldy  #2
 2059 b1 5f             lda  (tcim),y
 205b 65 5f             adc  tcim
 205d 85 47             sta  kov
 205f c8                iny
 2060 b1 5f             lda  (tcim),y
 2062 65 60             adc  tcim+1
 2064 85 48             sta  kov+1
 2066 a0 00             ldy  #0
 2068 a5 47     hasonl  lda  kov
 206a c5 31             cmp  tvege
 206c d0 06             bne  ciklus
 206e a5 48             lda  kov+1
 2070 c5 32             cmp  tvege+1
 2072 f0 13             beq  cikveg
 2074 b1 47     ciklus  lda  (kov),y
 2076 91 5f             sta  (tcim),y
 2078 e6 47             inc  kov
 207a d0 02             bne  f1
 207c e6 48             inc  kov+1
 207e e6 5f     f1      inc  tcim
 2080 d0 e6             bne  hasonl
 2082 e6 60             inc  tcim+1
 2084 4c 68 20          jmp  hasonl
 2087 a5 5f     cikveg  lda  tcim
 2089 85 31             sta  tvege

 208b a5 60             lda  tcim+1
 208d 85 32             sta  tvege+1
 208f 60                rts


2000-2090
```

3. ábra

```
100 let x=50
110 dim a$(x)
120 let i=1
130 do
140 print i;". nev ";
150 input b$
160 if b$="vege" then exit
170 let a$(i)=b$
180 let i=i+1
190 if i=x+1 then gosub 380:rem tombnov
200 loop
210 gosub 250:rem rendezes
220 gosub 310:rem kiiras
230 end
240 :
250 rem rendezo algoritmus
260 rem
270 rem
280 :
290 return
300 :
310 rem kiiro algoritmus
320 rem
330 rem
340 :
350 return
360 :
370 rem tombnoveles
380 dim s$(x)
390 for k=1 to x
400 let s$(k)=a$(k)
410 next k
420 erase a$(1)
430 let x=x+10
440 dim a$(x)
450 for k=1 to x-10
460 let a$(k)=s$(k)
470 next k
480 erase s$(1)
490 return
```

Hány színű egy térkép?

A térképek színezésével kapcsolatban egy klasszikus matematikai feladat megoldása igen sokáig váratott magára. A matematikának ilyen típusú problémákkal foglalkozó ágában, a topológiában járatosak számára régóta ismeretes volt az a sejtés, hogy minden térkép kiszínezhető legfeljebb négy színnel úgy, hogy az egymással határos országok különböző színűek legyenek.

A sejtés csak a közelmúltban vált bizonyossággá, amikor azt számítástechnikai támogatással igazolták. Mi nem a tétel bizonyításával — ami komoly matematikai apparátust igényel —, hanem csak alkalmazásával foglalkozunk.

A következő Turbo Pascal program egy adott térkép konkrét színezését végzi el rekurzív algoritmus szerint. A program a halmazok alkalmazását, valamint a táblázatba rendezett menük kezelését is bemutatja. Az itt közölt változatban a 30 európai országot építettük be, de ez más, tetszőleges térképre is átalakítható. Ehhez az országok számát *(n)*, valamint az *m1* tömbben tárolt országneveket kell módosítani.

Az országokat az 1, 2, . . . n számokkal kódoltuk, a határokat pedig úgy írjuk le, hogy minden országhoz megadjuk szomszédainak halmazát (ez egy országokkal indexelt, országokból álló halmazokat tartalmazó tömb, melynek neve *hataros*), végül a színezést szintén egy tömbbel fejezzük ki, amelynek indexei a lehetséges színek, elemei pedig az egyszínű országok halmazai (a tömb neve: *szinuek*).

```
program terkep;

{Térkép színezés 4 színnel, rekurzív algoritmussal}

const n = 30;

type orszag = 1..n;
     orszagok = set of orszag;
     szin = (lila, zold, sarga, piros);
     str20 = string[20];
     fun = (Eleje, Fel, PgUp,     {Vezérlő billentyűk a menükezeléshez}
            Bal,        Jobb,
            Vege,  Le,  PgDn,
            Ret,Semmi);
     strtomb = array[1..n] of str20;

var hataros: array[orszag] of orszagok;
             {Egy adott országgal határos országok halmazai}
    szinuek: array[szin] of orszagok;
             {A pillanatnyilag azonos színű országok halmazai}
    sz: szin;

const m1: strtomb = ('Albánia','Andorra','Ausztria',
                     'Belgium','Bulgária','Csehszlovákia',
                     'Dánia','Finnország','Franciaország',
                     'Görögország','Hollandia','Irország',
                     'Izland','Jugoszlávia','Lengyelország',
                     'Liechtenstein','Luxemburg','Magyarország',
                     'Nagybritannia','NDK','Norvégia',
                     'NSZK','Olaszország','Portugália',
                     'Románia','Spanyolország','Svájc',
                     'Svédország','Szovjetunió','Törökország');

procedure wait;
begin gotoxy(1,25); write('Folytatás tetszőleges billentyűre':56);
      repeat
      until keypressed;
      clrscr
end;

procedure menuset2d(var es: orszagok; n: integer; honnan: strtomb);

{1.2.1. eljárás általánosított változata, melyben a menüből egy halmaz
        válogatható ki}

{es:     Kiválasztott országok halmaza}
{n:      Menütételek száma}
{honnan: A menüszöveget tartalamazó strtomb neve}

{Választás RETURN -el, kilépés End -el}

var i, j, k, l, y: integer;
    c: fun;

procedure inverz;
begin textbackground(white);
      textcolor(black)
end;
```

A *hataros* tömb feltöltését a program végzi a felhasználó által szolgáltatott adatok alapján. Az adatbevitel a *menuset2d* eljárással történik, amely a menü legfeljebb 20 karakter hosszú tételeit táblázatos formában, soronként négyesével jeleníti meg a képernyőn. Választani (*Enterrel*) a tételek egy, esetleg üres halmazát lehet, választás után az End billentyűvel kell kilépni. Az eljárás az 1989/6. számban közölt *menuset* általánosított változata.

A program érdemi része tulajdonképpen három eljárásból áll.

hatarbe. Sorra veszi az országokat, és mindegyikhez beolvassa a vele szomszédos országok halmazát a *menuset2d* segítségével.

eredmeny. Az eredményt közli oly módon, hogy minden országot kikeres a *szinuek* tömbből, és kiírja a színével együtt. Az eljárás egy *halt* utasítással fejeződik be, mivel csak a program végén, egyszer kerül hívásra, és ezután már nem szabad a hívó programba visszatérni.

szinez. A program központi része, amely egy rekurzív algoritmus alapján a színezést végzi. Maga a színezés azt jelenti, hogy az országot csatlakoztatni kell a *szinuek* tömb valamelyik halmazához. Az algoritmus igen egyszerű: sorra vesszük a színeket, és minden színhez ellenőrizzük, hogy a kérdéses országhoz tartozó *hataros* tömbbeli halmaznak és az aktuális színű országoknak (ez a *szinuek* tömb megfelelő eleme) van-e közös része. Ha van, színt kell váltani, ha nincs, az ország megkapja az aktuális színt, és megnézzük, hogy az utolsó országot színeztük-e. Ha nem, az eljárás önmagát hívja a következő ország színezésére. Kilépni csak akkor lehet, ha az utolsó országot is kiszíneztük (ekkor hívjuk az *eredmeny* eljárást), vagy ha a színek elfogytak. Ez utóbbi esetben az előző rekurzív szintre kerülünk vissza, ahol megszüntetjük a korábban elvégzett, de zsákutcának bizonyult színezést (a *szinuek* tömb megfelelő eleméből kivesszük az országot). Ha a feladatnak van megoldása — és ezt most már egy matematikai tétel garantálja —, a főprogramba nem is térünk vissza, mert az *eredmeny* egy *halt* utasítással végződik.

Figyeljük meg, hogy a program logikailag bonyolultabb részei — a *szinez* eljárás és a főprogram — milyen rövidek a menükezeléshez és az adatbevitelhez képest. A főprogramra csupán a *szinuek* tömb kezdeti törlése és a „lavina" elindítása marad az első ország kiszínezésével. Mint látható, a rekurzív algoritmus meglehetősen tömör és elegáns programhoz vezet, melynek megértése azonban nem mindenkinek egyszerű.

Bakos Tamás

```
procedure normal;
begin textbackground(black);
      textcolor(white)
end;

function funkod: fun;

var c: char;
begin read(kbd,c);
      if c=#13
      then funkod:=Ret
      else if (c=#27) and keypressed
           then begin read(kbd,c);
                      case c of
                           #71: funkod:=Eleje;
                           #72: funkod:=Fel;
                           #73: funkod:=PgUp;
                           #75: funkod:=Bal;
                           #77: funkod:=Jobb;
                           #79: funkod:=Vege;
                           #80: funkod:=Le;
                           #81: funkod:=PgDn;
                      else funkod:=Semmi
                      end
                end
           else funkod:=Semmi
end;

{menuset2d törzse következik:}

begin i:=n div 4; l:=1; y:=wherey; gotoxy(1,y+2);
      for j:=1 to i do
          begin for k:=1 to 4 do  {a menü tételek kiírása}
                     begin write(honnan[l]:20);
                           l:=l+1   {4 ország soronként}
                     end;
                writeln
          end;
      for k:=l to n do
          write(honnan[k]:20); {az esetleges maradék kiírása}
      inverz;
      gotoxy(1,y+2); write(honnan[1]:20);
      normal; i:=0; j:=0; es:=[];
      repeat c:=funkod
      until c in [Ret, Le, Fel, Bal, Jobb, Vege];
      while (c<>Vege) do    {4*i+j+1 = az i. sor j. elemének lineáris}
            begin if (c<>Ret) and (not(4*i+j+1 in es))        {indexe}
                  then begin gotoxy(1+20*j,y+2*i+2);
                             write(honnan[4*i+j+1]:20)
                       end;
                  case c of
                       Le:    if (i<n div 4) and (4*i+j+1<=n-4)
                              then i:=i+1;
                                        {Navigálás a menüben.}
                       Fel:   if i>=1  {a sorok és oszlopok vége nem}
                              then i:=i-1;  {csatlakozik az elejéhez}

                       Bal:   if j>=1
                              then j:=j-1;
```

```
                    Jobb:  if (j<3) and (4*i+j+1<n)
                           then j:=j+1;

                    Ret:   es:=es+[4*i+j+1]      {Választás Ret -el}
               end;
               inverz;
               gotoxy(1+20*j,y+2*i+2); write(honnan[4*i+j+1]:20);
               normal;
               repeat c:=funkod
               until c in [Ret, Le, Fel, Jobb, Bal, Vege]
          end
end;

procedure hatarbe; {A "hataros" tömb kitöltése}
var i: orszag; {Minden országhoz a vele határos országok halmaza
                                                      tartozik}

begin for i:=1 to n do
       begin clrscr; write('Az ország neve: ',m1[i], ',szomszédai:');
             menuset2d(hataros[i],n,m1)
       end
end;

procedure eredmeny;  {Az elkészült színezés kiírása}
var i: orszag;
    c: szin;

begin clrscr; writeln('Az országok szinezése a következő':56);
      Writeln('--------------------------------':56);
      for i:=1 to n do
          begin write(m1[i]:20,'   '); c:=lila;
                while not(i in szinuek[c]) do
                      c:=succ(c);              {A szín kikeresése}
                case c of
                     lila:  writeln('LILA');  {és kiírása}
                     zold:  writeln('ZÖLD');
                     sarga: writeln('SÁRGA');
                     piros: writeln('PIROS')
                end;
                if (i mod 20)=0      {20 soronként szünet}
                then wait
          end;
      halt     {A kiírás végén az egész programból ki kell lépni}
end;

procedure szinez(i: orszag);
var c: szin;                  {Az "i" nev" ország kiszínezése}

begin for c:=lila to piros do
          if hataros[i]*szinuek[c]=[]  {i szomszédai között nem lehet
                                                        c szín"}
          then begin szinuek[c]:=szinuek[c]+[i];     {legyen i c szín"}
                     if i=n      {ha ez volt az utolsó, készen vagyunk}
                     then eredmeny
                     else szinez(i+1);     {különben a következőt kell
                                                        színezni}
                     szinuek[c]:=szinuek[c]-[i]  {Nem sikerült i=n-ig}
               end             {eljutni, i -től el kell venni a c színt}
end;



{főprogram törzse:}
begin hatarbe;
      for sz:=lila to piros do     {az egyszínek halmazainak törlése}
          szinuek[sz]:=[];
      szinez(1);                    {itt indul a színezés}
      write('Nincs megoldás!')   {Ha nem az "eredmeny" -ből lépünk ki}
end.
```

Adom a magyarázatot!

Adom a magyarázatot az 1989/8-as számban megjelent problémákra.

1. PRINT 1+"2"+—1 . . . lefagyás

A hiba oka: az operációs rendszer mindent ugyanazzal a beolvasó rutinnal olvas be, így a mínusz jelet előjelcsereként értelmezi és elkavarodik. Az operációs rendszer a füzérek összeadásáig ($B63D) minden baj nélkül eljut. Elmenti az első füzér azonosítóját a verembe ($01,19), majd jön a második füzér, amit az általános beolvasó rutinnal szeretne beolvasni, de ott a mínusz jel, és így eljut $AF0D-re, ahol elköveti az első fatális hibát: kiveszi a veremből a visszatérési címet (most $0019 van a verem tetején).

```
$AF0D LDY # 15
$AF0F PLA
$AF10 PLA
$AF11 JMP $ADFA
```

```
$ADF9 PHA
$ADFA JSR $AE20
$ADFD PLA
```

A következő JMP egy PHA—PLA szerkezet közepébe ugrik, ez később hibához vezet. Először megcsinálja az előjelcserét ($AE20), majd jön az a bizonyos PLA (a verem tetején $0019 van) egy sor rossz következtetés, és a gép újabb bájtokat vesz ki a veremből, és amikor jön az RTS, akkor a veremben az első numerikus adat mantissza-darabja van (ebben az esetben $0000) . . . kész a fagyás. A gép a RUN/STOP-RESTORE kombinációra azért nem reagál, mert közben CRASH utasításra futott (egy illegális kód).

Mivel a verem tetején az első szám egy darabja van, ezért ügyes számkombinációkkal bármilyen rutint meg lehet hívni egy BASIC PRINT paranccsal. Például:

PRINT —0.311+"2"+—1 Video reset.

LOAD (majd RUN/STOP) és ezután

PRINT —32.145033795+"2"+—1 (ez egy újabb LOAD utasítást eredményez)

X=—0.987444579, majd PRINT X+"2"+—1 (hidegindítás)

Ezzel a trükkel új lehetőségek nyílnak a BASIC programok titkosításában. Ki veszi észre például, hogy egy PRINT utasítás hidegindítást csinál?

2. 10 LOAD "S", 8 majd RUN, és ezután a nulladik sor törlése „elszálláshoz" vezet.

A probléma nagyon egyszerű. Ha programból adjuk ki a LOAD parancsot, akkor az operációs rendszer nem állítja be a végcímmutatókat. Ha ilyenkor javítunk a programunkon, akkor a beszúró-törlő rutin rosszul működik.

Újlaky Attila

Programozási fogások és melléfogások

A Mikroszámítógép Magazin 1988/10. számában *Kivétel erősíti a szabályt* közös cím alatt három, játékos jellegű program listája jelent meg. Egy mondat a beharangozó szövegből: „Ezúttal mégis közreadunk három különböző géptípusra írt programot, mert úgy véljük, hogy egyrészt szórakoztatóak, másrészt *programozástechnikai fogásokat* leshetünk el belőlük". Érdeklődésemet felkeltette a Plus/4-re írt BIORITMUS program, melyről rövid idő alatt megállapítottam, hogy a programozási *melléfogások* hemzsegnek benne. Az alábbiakban ezekből mutatok be néhányat.

A mellékelt *listán* a programnak csak egyes részleteit idézem, jó néhány hiba szinte gépiesen ismétlődik, más esetekben elegendőnek vélem a szavakkal történő leírást. Aki a részletekre kíváncsi, könnyen utánanézhet. Korábbi szokásomtól eltérően a javításokra is csak szöveges utalást teszek, programlista nélkül. A teljes program javításával kár lenne kísérletezni, mert alapvető hibái miatt egyszerűbb egy teljesen új programot tervezni és megírni.

A program egyetlen erénye, hogy egyenletesen, 10-esével van sorszámozva, ami az AUTO parancs használatával megkönnyíti a begépelést. Sajnos ezzel nem sokra megyünk, hiszen még több munkát takaríthatunk meg, ha a programot egyáltalán nem írjuk be, ugyanis az általa nyújtott információk *100 százalékig használhatatlanok*. Ezen még egy BASIC ellenőrzőkód sem segítene, legfeljebb a tapasztalatlan olvasót nyugtatná meg, hogy nem ő követett el gépelési hibát.

A program mérete (3714 bájt) nem tette lehetővé, hogy C16-on is futtatni tudjam, ezért csupán az alkalmazott BASIC-verzió ismeretére és többéves programozói tapasztalataimra tudtam támaszkodni. Kételkedem abban, hogy ezt a programot közlés előtt bárki hozzáértő módon kipróbálta volna, természetesen beleértve a szerzőt is.

Ezek után nézzük először a listát. Indulásnál töröljük a képernyőt. Kétszer is: előbb a GRAPHIC utasítás második operandusával, rögtön utána az SCNCLR utasítással. Nem nagy hiba, hisz az a biztos, amit az ember kétszer mond, kétszer mond.

A 350-es sorban látható szerkezet már többször szerepelt ebben a sorozatban. Ha a vizsgálat feltételét megfordítjuk (A$ < > "I"), az ELSE ág feleslegessé válik, elhagyható.

A 420-as sorban definiáljuk a 12 elemű H vektort, majd feltöltjük a hónapok napjainak a számával. A READ-hez tartozó DATA-sorokat egy kissé távolabb, a 680-as sortól kezdve helyezzük el, egyrészt azért, hogy ne lehessen olyan könnyen megtalálni, másrészt a számítási eljárások közé beépítve egy ügyes trükkel megoldható azok szédületes sebességének csökkentése. Ha GOSUB 700 helyett GOSUB 680-at írunk, az interpreternek minden egyes alkalommal *karakterenként* át kell rágnia magát a DATA-utasításokon. Ez nagyon hatékony módszer a lassításra, különösen ciklusban, s az 570-es sor láttán meg is győződhetünk arról, hogy valóban ciklusról van szó. Az itt közölt listán nem látható, hogy az egész programban öt helyen fordul elő a GOSUB 680 utasítás, összesen három esetben ciklikusan.

Miután a H vektorba beolvastuk a szükséges adatokat, egy CLR utasítással gondosan töröljük a változókat, köztük az imént definiált H vektort is. Ez a 730-as sorban a B változó 10-nél nagyobb értéke esetén *BAD SUBSCRIPT* hibaüzenethez vezetne, de a program készítője egy zseniális fogással elhárítja ezt a lehetőséget. A listán is látható, hogy a gondot okozó utasítás csak GOSUB 730, esetleg GOTO 730 utasítással lenne elérhető; az eredeti program gondos átbogarászásával sem bukkanunk ilyen utasításra.

Az 500-as soron kezdődő bonyolult számítási algoritmust sajnos hiába próbáltam elemezni, többször nekifutásra sem sikerült. Maga az 500-as sor felesleges, mert az értékadó utasítások bal oldalain szereplő A2, B2 és C2 változók a programban többször nem fordulnak elő.

Az 580-as sor második utasítása szintén ismerős a sorozat olvasói számára: ugrás a GOTO-ra. A GOTO 650 más helyen is előfordul.

Bioritmus-programokra jellemző hiba, hogy figyelmen kívül hagyják a szökőnapokat. Itt erre egy teljesen eredeti megoldást láthatunk a 660—670-es sorokban. Az itt látható algoritmus lehetővé teszi, hogy minden évet szökőévként kezeljünk. Ugyanis a szubrutinból való visszatérésnél I értéke mindig 0 lesz, ami a néggyel maradék nélkül osztható évszámot, vagyis szökőévet jelent. Ha a 660-as sor elején I=INT (A/4) állna, akkor már nem lenne ilyen egyértelmű a dolog.

A 710-es és 720-as sorokat egy sorrá össze lehetne vonni, például az N=N+29—I—K értékadás használatával. Csak az a baj, hogy az ilyen fogásokkal a program mérete csökken, és félő, hogy a begépelést vállaló naiv olvasó ellustul.

Szerencsére vannak még jó módszerek az ellustulás ellen. Nem szabad a sok azonos részletet tartalmazó rajzoló eljárásokat — paraméterátadással működő — szubrutinba tenni, mert akkor jelentősen csökkenne a program mérete. A dolgot tovább lehet cifrázni az 1170—1180-as sorokban látható fogással, amely — az előző mondatban leírtak szellemében — szintén háromszor fordul elő. Bár GETKEY után nem lehet A$ értéke üres sztring, de ha az lenne, akkor is ugyanott folytatódna a futás, ahol anélkül. Fantasztikus!

Nem utolsó dolog — bár a *CAD* című könyvről szóló részben már láttunk hasonlót — az 1300-as sor E=E értékadása, amelyhez hasonló szintén van még kettő.

Sose jutna magamtól eszembe, hogy egy *pontot* az 1310-es sorban látható módon rajzoljak a képernyőre. Megelégednék a DRAW, W, Y utasítással, ha ez a program nem sulykolja belém ezt az igazán üdvözítő módszert.

Bevallom, nekem a program utolsó utasítása tetszett a legjobban: SYS 32768 — BASIC hidegindítás. Ezzel kellett volna kezdeni. Sokkal jobb, mint a hideg zuhany és a kényszerzubbony. Pedig ha sok hasonló programot látok, előbb-utóbb azt sem kerülöm el.

Barna László

```
10 GRAPHIC3,1:SCNCLR:VOL6
   ...
340 GETKEYA$
350 IFA$="I"THEN360:ELSE250
360 GRAPHIC0:SCNCLR:VOL7
   ...
420 DIMH(12):FORI=1TO12:READH(I):NEXT
430 CLR:N=0:PRINT"(CLR)(4 DOWN)";
   ...
500 A2=A:B2=B:C2=C
   ...
560 K=C:GOSUB580:K=0
570 B=B+1:IFB<L THEN GOSUB680:GOTO570
580 N=N+Z:GOTO650
   ...
650 GOTO760
660 I=A/4:IFI*4=A THENI=0:ELSEI=1
670 RETURN
680 DATA 31,28,31,30,31,30
690 DATA 31,31,30,31,30,31
700 IFB=2 THENJ=A:GOSUB660
710 IFI=0 THENN=N+29-K:RETURN
720 N=N+28-K:RETURN
730 N=N+H(B)-K:RETURN
   ...
1170 GETKEYA$
1180 IFA$=""THEN1190
1190 SCNCLR:VOL5
   ...
1290 FORY=50TO160
1300 E=E:W=E*4
1310 DRAW,W,YTOW,Y:NEXT
   ...
1770 SYS32768
```

Válaszoltak a tulajdonosok

A Magazin előző számaiban megjelent az Enterprise-TOTÓ és az információs lap. Ezek eredményeiről kérdeztük Pásztor Tamást, a Centrum Áruházak Vállalat márkamenedzserét. Az ő értékelését adjuk közre az alábbiakban.

Várakozáson felüli érdeklődés

A TOTÓ I-re több mint 1700 válasz érkezett. Ez a szám az Enterprise-tulajdonosok tíz százalékát jelenti, az érdeklődés tehát várakozáson felüli volt. Nem titkolt szándékunk volt, hogy a TOTÓ-játékon keresztül is információkat gyűjtsünk az Enterprise-osok táboráról. Ezért is örülünk a kedvező fogadtatásnak.

Az információs lapra szeptember elejéig 1550 válasz érkezett. Ez azonban még nem végleges szám, mert továbbra is folyamatosan érkeznek adatlapok. A TOTÓ-szelvényeket és az információs lapot is számítógéppel dolgoztuk fel, mert össze akarjuk gyűjteni az Enterprise-tulajdonosok címeit. Adatbankunkban eddig mintegy 2500 cím szerepel. El szeretném mondani, hogy a személyi számokat csak az adatfeldolgozás miatt, technikai okokból kértük, ezeket az információkat később meg fogjuk semmisíteni. Sok választ kaptunk Csehszlovákiából és a Szovjetunióból. Ez megerősítette azt az eddigi érzésünket, hogy nagyon sok gépet vásároltak határmenti forgalomban.

A számítógépes feldolgozás néhány érdekes dolgot is „kihozott". Néhány versenyző több, egymástól eltérő választ küldött be. Ez azt jelenti, hogy nem tudták a jó választ, és azok variálásával akartak jó eredményt elérni. Ez ötletes fogás ugyan, de nem tartjuk tisztességesnek, mivel nem a tényleges ismereteket tükrözi. Így hosszas megfontolás után a versenybizottság kizárta a versenyből Bodnár Zoltánt (Kazincbarcika, 14 db szelvény), Bujtor Gyulát (Budapest, 7 db szelvény) és Kőhalmi Attilát (Ajka, 5 db szelvény). Néhány versenyző a szerencséjét a beküldött szelvények számával próbálta meg elősegíteni. Őket a versenybizottság intésben részesítette, de nem zárta ki.

Összesen két darab 13+1-es és öt 13-as találatot elért szelvény érkezett. Spectrum-emulátort Fehér Péter (Pécs), EP-egeret Fehér Katalin (Kaposvár) nyert. Akik 13 találatos szelvényt küldtek be, tíz játékprogramos kazettát kapnak. Ők a következők: Fulup Csaba (Debrecen), Nagy Krisztián (Nyírbátor), Földes Imre (Budapest), Fisli Gyula (Kaposvár), Lengyel Zsolt (Kecskemét).

Sajnálatosan mind a két TOTÓ-ba került nyomdahiba. Az elsőben a NASA-GAY program neve hibásan jelent meg, ez azonban nem volt zavaró. A TOTÓ II-ben viszont a 7. kérdést törölni kellett, mivel kimaradt a „nem" szó, és ez már félreértésre adhatott okot. A kérdés helyesen: A felsorolt programok közül melyik *nem* jelent meg áruházi forgalomban?

Le kell vonni a tanulságokat!

Említettem, hogy az Enterprise-piaccal kapcsolatos kérdésekre is igen sok választ kaptunk. Ez szignifikáns minta, és komolyan kell vennünk a válaszokat. A válaszok 30 százaléka Pestről, 70 százaléka vidékről érkezett. Ez már önmagában is megdöbbentő adat, mert a gépeket éppen fordított arányban adtuk el. A gépek típusának megoszlása is ellentétes az eladásokkal. A beküldők megoszlásából arra kell következtetnünk, hogy a gépnek vidéken nincs supportja. Néhány áruház tartja csak magát, mint a győri, miskolci, kecskeméti és a szegedi.

Továbbra is másolás

A szoftverre vonatkozó válaszokból egyértelműen kiderül, hogy gyenge az ellátás. A megkérdezetteknek átlagosan 9 gyári és 73 másolt játékprogramjuk van. A felhasználói programok száma 16, a könyveké 4. Ez a szám híven tükrözi a könyvpiacot. A programok sikerlistája a következőképpen alakult:

Játékok
1. Last Ninja II. (Spectrum-átirat)
2. BATMAN
3. Magic Ball

Felhasználói
1. ASMON/SIMON Assembler Compiler
2. Zzzip BASIC Compiler
3. Hisoft DEVPACK

Kiosztottuk a citromdíjat is. Ezt a következők kapták:
1. Körmöci arany
2. Bioritmus
3. CBM MFT (Commodore-segédprogram)

A hardver sikeresebb

A kiértékelés szerint a hardverrel kapcsolatban kevesebb gond van. A gép leggyengébb elemei: a fóliataszatúra, a billentyűzet és a csatlakozópontok. Sok kritika érte az adatmagnót és a system bus bridge-et. A gép leggyengébb szoftvereleme egybehangzóan a szövegszerkesztő.

Meglepő választ kaptunk arra a kérdésre, hogy: mivel bővítené a hardver-termékskálát? Az igények 7 százaléka a fényceruza. Ezt túlzott elvárásnak érezzük, mert a fényceruza függvénye a monitornak. A sorrend egyébként így alakult: numerikus tasztatúra, memóriabővítő, EPROM-égető, floppydoboz, midi interfész. A szoftvertermékek körét a jó szövegszerkesztővel, zenei programkészítővel, adatbázis-kezelővel, játékprogram-generátorral és oktatóprogramokkal bővítenék a válaszolók.

A kereskedelemre vonatkozó kérdésekre a következőképpen alakultak a válaszok:

A legjobban ellátott forgalmazó helyek:
1. Úttörő Áruház 29%
2. Novotrade 2C 11%
3. Centrum Flórián 6%

Egyéb vagy nem minősített 54%.
A legképzettebb eladókra vonatkozó válasz:
1. Úttörő Áruház 20%
2. Otthon Áruház 13%
3. Novotrade 2C 6%.
Egyéb, nem minősített 61%

Tervek és tanulságok

Az igényeket mérlegelni kell. Nem lehet mindenbe belevágni. Nem fogunk tudni lépni, beszerzési lehetőség hiányában, az olcsó nyomtató és színes monitor területén. Floppydobozt és tápegységet a közeljövőben fogunk árulni. Memóriabővítés esetében csak az alaplemezt forgalmazzuk, és a szerelést vállaljuk. A memória IC-ket a vásárló maga választhatja, pénztárcája függvényében. A beszerzéshez azonban segítséget nyújtunk.

Ami a szoftvert illeti, itt már keményebb kritikát kaptunk. A választék valóban gyenge. Főleg az alkalmazói programokat hiányolják, ami azért egyben igazolja előzetes elképzeléseinket is. Ugyanis mi az Enterprise-t családi gépnek szántuk. Ezért az alkalmazói programok fejlesztésére fogunk koncentrálni. Megfontoljuk a CP/M alatt futó programok fejlesztését, illetve a már meglévők jogdíjának megszerzését.

Nagyon elgondolkoztató, hogy a válaszolók három százaléka olyan programokat is keresett, amelyek léteznek és vannak is. Ez a forgalmazásunk kritikája! Különösen a vidéki felhasználókat sajnálom nagyon.

Pinke György

Az Enterprise-TOTÓ II. fordulójára beérkezett pályázatok száma közel ezer volt. A pályázatot kiíró Centrum Nagykereskedelmi Vállalat szeptember 30-ig elfogadta a beérkező tippeket.

Fontos! Szedési hiba miatt (a kérdésből kimaradt a nem szócska) törölni kellett a TOTÓ-ban szereplő 7. kérdést, gyakorlatilag tehát csak 13 kérdésre lehetett jó választ adni.

Sajnos a válaszok közlésénél is volt egy hiba, a 12. kérdésre adandó helyes válasz 1 lett volna a közölt 2 helyett —, mivel a Last Ninja II. Enterprise-os verziójának szintszáma: hat.

Tizenhárom találatos szelvény nem akadt az értékelés során. Tizenkét találatot ért el Szabó Attila budapesti olvasónk, így ő nyert egy Enterprise-egeret. Tizenegy találatos szelvényével 10—10 új Enterprise-programot nyert: Kiss Raffael (Budapest), a Matlári család (Budapest), Majdán József (Budapest), Borbás Tibor (Eger) és Heilig Szabolcs (Veszprém).

Mivel nem akadt olyan szelvény, amely az első és második fordulóban is 13+1, illetve 13 találatot ért volna el, a meghirdetett RX-80-as nyomtatót nem tudjuk kisorsolni.

Még most is folyamatosan érkeznek az INFO-kártyák, ezért a vigaszdíj sorsolását november elejére tettük el, eredményéről egy későbbi számunkban adunk hírt.

Az első és második fordulóban nyertes pályázók, valamint a vigaszdíj nyertesei egyszerre, még a karácsonyi vásár előtt magkapják nyereményeiket a Centrum Úttörő Áruházban, egy távirati kiértesítést követően.

Videomonitorként Junoszty

A cikk témáját rendkívül fontosnak tartjuk, de sajnos az átalakítás leírását kissé elnagyoltnak. Mégis úgy döntöttünk, hogy leközöljük Molnár Ferenc írását, mivel e monitorínséges időben segítséget nyújthat. Természetesen alaposabb leírásokat is várunk.

Az Enterprise ISDOS-lehetőségeinek kihasználásához már nem elegendő a normál televízió nyújtotta, 40 karakternek megfelelő felbontási képesség. Az 1989/3. számban egy példát olvastam a Junoszty-402 BC tévé videomonitorként történő alkalmazására. Azonnal szereztem egy használt készüléket, és hozzáláttam az átalakításhoz. A hang erősítéséről lemondtam, mert a gép saját hangját elegendőnek tartottam. A videojelnél a relés megszakítást bonyolultnak ítéltem, és helyette egy 3,5-es sztereo leválasztós jack aljzatot szereltem a fejhallgató-kivezetés helyére. A monitorcsatlakozó bedugása mindkét ágat megszakítja.

Ezután kezdődött a problémám, hogy az Enterprise-monitor aljzatáról melyik kimenetet vagy kimeneteket kössem össze a monitorommal. Hosszas kísérletezés után elfogadható képet csak úgy kaptam, ha a számítógép monitorkimenetén a 6-os üzemmódkapcsolót és a C összetett videojelet közös vezetéken, illetve a testkivezetést csatlakoztattam a Junoszty monitorhoz. Be kell vallanom, hogy kezdetben nagy volt az örömöm, mert a 80 karakteres képernyőn is jól olvasható, szemet pihentető, jó minőségű képet kaptam. Később észrevettem, hogy mégsem minden tökéletes. A piros és a zöld színű betűk teljesen azonos árnyalatúak. (A tévékimenetre csatlakoztatva ez a két szín élesen elválik fekete-fehér képernyőn is.) Egyes játékprogramok emiatt élvezhetetlenek. Továbbá a képernyőn a kontraszt lényegesen lágyabb, mint a tévékimenetről, bár ezen hangolással lehet segíteni.

Összességében a Junosztyot mint videomonitort hasznosnak tartom az Enterprise-hoz, még a fent leírt kifogások ellenére is. Az átalakítás óta a számítógéphez nem használtam színes tévét, mert kevesebb élményt nyújtott. A családdal is állandóan meg kellett állapodni a tévé használatában. Azóta a számítógép-konfigurációt nem kell megbontanom. Külön öröm volna számomra, ha valaki hasznos tanácsot tudna adni a leírt probléma megoldására is.

Molnár Ferenc

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub
a VSZM
Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 181-0950/473

Az Enterprise nyolc királynője

Az alábbi cikk eredetileg Pintér Gábor *Feladatok — megoldások* sorozatának 14. feladatához kapcsolódik. Mivel magazinunk régi ismerőse, *Kóta Béla*, ízigvérig Enterprise-os, úgy döntöttünk, hogy írásának ebben a rovatban van a helye.

Pintér Gábor szerint a kiindulásnál a sakktábla minden oszlopában egy királynő állhat. Ezzel a korláttal a lehetséges eseteket jelentősen leszűkíti, majd egy léptető és ellenőrző algoritmussal vizsgálja, hogy az egymás után a táblára tett királynő üti-e a táblán lévőket. A továbbiakban egy vagy sikerül, vagy nem eljárást alkalmaz.

Felhívom figyelmüket arra, hogy a feltételeknek megfelelően *egy sorban és egy oszlopban is csak egy királynő állhat!* Ennek alapján a célszerű kiindulásnál a 8 királynő álljon a tábla átlójában. Saját programomban egy N=4 . . . 8 elemű MZ$ tömb jelképezi a kiinduló tábla sorait. A sorok cserélgetésével lehet megkeresni a megfelelő elrendezéseket. A kiinduló forrássorrendet egy X$="1234 . . . N" sztring jelképezi. A végcélsorrend G$="N . . . 4321". (A V$, H$ a tábla oszlop- és sorfelirata.)

A matematikailag korrekt eljárás így a 8 elem minden lehetséges kombinációjának vizsgálata. A kombinációk száma így csak max. 40320=8! (lásd az *1. listát*).

A ZZZIP-pel lefordított program 8×8-as táblán a lehetséges 92 megoldást 7 perc 16 másodperc alatt számítja ki és rajzolja meg *(2. lista)*. A programot átnyomorítottam Microsoft GWBASIC-re is. BASCOM fordítóval lefordítva, PC/XT gépen turbóval a futásidő 5 perc 30 másodperc.

1. lista

```
A kiindulá változók felépítése:

  250 LET X#,V#,H#,G#=""
  260 FOR I=1 TO N
  270   LET MZ#(I)="":LET X#=X#&CHR#
(I+48):LET V#=V#&CHR#(I+48):LET G#=G#&
CHR#(N+49-I):LET H#=H#&CHR#(I+64)
  280 NEXT I
  290 LET H#=" "&H#:LET J#="r"
  300 !
  310 FOR I=1 TO N
  320   FOR J=1 TO I-1
  330     LET MZ#(I)=MZ#(I)&CHR#(128)
  340   NEXT J
  350   LET MZ#(I)=MZ#(I)&CHR#(142)
  360   FOR J=I+1 TO N
  370     LET MZ#(I)=MZ#(I)&CHR#(128)
  380   NEXT J
  390 NEXT I
  ...
  410 CALL MAIN
  ...
  430 END
A MAIN függvény állítja elő az elemek
összes kombinációját - EGYSZER!

  450 DEF MAIN
  ...
  470   DO
  ...
  510     CALL UTES
  520     FOR I=N-1 TO 1 STEP-1
  530       IF X#(I)<X#(I+1) THEN EXIT
FOR
  540     NEXT I
  550     FOR J=N TO 1 STEP-1
  560       IF X#(J)>X#(I) THEN EXIT
FOR
  570     NEXT J
  580     LET C#=X#(I+1:J-1)&X#(I)&X#
(J+1:):LET X#=X#(:I-1)&X#(J)
  590     FOR J=LEN(C#) TO 1 STEP-1
  600       LET X#=X#&C#(J)
  610     NEXT J
  620   LOOP UNTIL X#=G#
  630 END DEF

Az UTES függvény választja ki a
megfelelő állásokat. U=N+2

  650 DEF UTES
  660   FOR A=1 TO N-1
  670     LET C=ORD(X#(A))
  680     FOR B=A+1 TO N
  690       IF B-A=ABS(ORD(X#(B))-C)
THEN LET A,B=U
  700     NEXT B
  710   NEXT A
  720   IF A<U THEN CALL ERED#émy
  730 END DEF
  740 !
```

2. lista

```
  100 PROGRAM "tape-2:green.bas"
  110 STRING MZ$(0 TO 8)
  120 SET BORDER 248:SET #102:PALETTE 7,7,7,7:SET #102:PAPER 0:SET #102:INK 1:
DISPLAY TEXT:CLEAR TEXT
  130 SET CHARACTER 128,255,129,129,129,129,129,129,255
  140 PRINT AT 10,1:"";:PRINT "Nyolc királynő";CHR$(241);PRINT "
----------";CHR$(241):PRINT "ENTERPRISE-128/ID-BASIC/ZZZIP";CHR$(241):
PRINT
  ...
  160 PRINT :SET #102:PAPER 2:SET #102:INK 3:PRINT "rajz be - R, ki - SPACE":
SET #102:PAPER 0:SET #102:INK 1:PRINT
  170 SET BORDER 248:SET #102:PALETTE 7,0,199,56
  180 DO
  190   INPUT PROMPT "Tabla meret (4+4-8+8) N = ":N
  200 LOOP UNTIL N>3 AND N<9
  210 LET U=N+2
  220 LET CH=2:SET VIDEO X N+2:SET VIDEO Y 2*N+6:SET VIDEO MODE 0:SET VIDEO
COLOUR 0:OPEN #CH:"video:"
  230 SET #CH:PALETTE 199,0,199,199:SET #CH:PAPER 0:SET #102:INK 1:DISPLAY #
CH:AT 1 FROM 1 TO 2*N+5
  240 !
  250 LET X$,V$,H$,G$=""
  260 FOR I=1 TO N
  270   LET MZ$(I)="":LET X$=X$&CHR$(I+48):LET V$=V$&CHR$(I+48):LET G$=G$&CHR$
(N+49-I):LET H$=H$&CHR$(I+64)
  280 NEXT I
  290 LET H$=" "&H$:LET J$="r"
  300 !
  310 FOR I=1 TO N
  320   FOR J=1 TO I-1
  330     LET MZ$(I)=MZ$(I)&CHR$(128)
  340   NEXT J
  350   LET MZ$(I)=MZ$(I)&CHR$(142)
  360   FOR J=I+1 TO N
  370     LET MZ$(I)=MZ$(I)&CHR$(128)
  380   NEXT J
  390 NEXT I
  400 PRINT #0:I$
  410 CALL MAIN
  420 DISPLAY TEXT:PRINT I$:PRINT "TIME=";TIME$
  430 END
  440 !
  450 DEF MAIN
  460   LET TL=0:TIME "00:00:00"
  470   DO
  480     LET I$=INKEY$
  490     IF I$<>"" THEN LET J$=I$
  500     IF J$="r" THEN CALL RAJZ
  510     CALL UTES
  520     FOR I=N-1 TO 1 STEP-1
  530       IF X$(I)<I$(I+1) THEN EXIT FOR
  540     NEXT I
  550     FOR J=N TO 1 STEP-1
  560       IF X$(J)>X$(I) THEN EXIT FOR
  570     NEXT J
  580     LET C$=X$(I+1:J-1)&X$(I)&X$(J+1:):LET X$=X$(:I-1)&X$(J)
  590     FOR J=LEN(C$) TO 1 STEP-1
  600       LET X$=X$&C$(J)
  610     NEXT J
  620   LOOP UNTIL X$=G$
  630 END DEF
  640 !
  650 DEF UTES
  660   FOR A=1 TO N-1
  670     LET C=ORD(X$(A))
  680     FOR B=A+1 TO N
  690       IF B-A=ABS(ORD(X$(B))-C) THEN LET A,B=U
  700     NEXT B
  710   NEXT A
  720   IF A<U THEN CALL ERED
  730 END DEF
  740 !
  750 DEF ERED
  760   LET TL=TL+1:PING
  770   CALL RAJZ
  780   PRINT #CH,AT N+5,1:"";
  790   CALL TABLA
  800   PRINT #0:I$;" -";TL
  810 END DEF
  820 !
  830 DEF RAJZ
  840   PRINT #CH,AT 2,1:"";
  850   CALL TABLA
  860   PRINT #CH:" ";X$
  870 END DEF
  880 !
  890 DEF TABLA
  900   FOR RZ=N TO 1 STEP-1
  910     PRINT #CH:V$(RZ):MZ$(VAL(X$(RZ)))
  920   NEXT RZ
  930   PRINT #CH:H$
  940 END DEF
```

Mi a manó?

A Mi a manó? ezúttal a szokásosnál nagyobb terjedelemben jelentkezik. No, nem azért, mert ez a karácsonyi szám, hanem mert dr. Nagy Zoltán tanácsait nem akartuk részletekben közölni. Köszönjük levelét és várjuk — nemcsak a programvédelemhez — a további hozzászólásokat.

A Mikroszámítógép Magazin 1989/8. számában a *Mi a manó?* rovatban vetődött fel a BASIC programok listázás elleni védelmének kérdése. A teljes védelemre én sem tudok megoldást, de a kíváncsi természetűek dolgát jelentősen meg lehet nehezíteni oly módon, hogy a program sorainak listázása csak egyenként, külön utasításokkal lehetséges, és az a 30—40-edik sornál már erősen fárasztó.

Tennivalónk a következő. Programunk minden sorának végére megjegyzésként egy tetszőleges karaktert kell beírni *(1. program)*.

1. program

```
100 PROGRAM PELDA !a
110 NUMERIC SZAM(11,2) !a
120 FOR A=1 TO 11 !a
130 FOR B=1 TO 2 !a
140 LET SZAM(A,B)=0 !a
150 NEXT B !a
160 NEXT A !a
170 ! és így tovább
.
.
.
```

Ha programunk készen van, az edit 1 paranccsal váltsunk programot, és írjuk be a *2. programot:*

2. program

```
100 LET N=4809 !angol billentyűzetű gép első sorának hossz-
                bájtja
110 DO
120 LET N=N+PEEK(N) !a következő sor hosszbájtja
130 POKE N-2,0
140 LOOP UNTIL PEEK(N)=0
150 END
```

Futtassuk le, majd edit 0-val váltsunk vissza eredeti programunkra, és próbáljuk meg kilistázni. Mint látjuk, a listázás az első sornál megáll. A LIST sorszám TO sorszám utasítás után is csak egy sor jelenik meg. Ennek ellenére programunk egyéb tulajdonságai nem változnak.

A módszer lényege, hogy a gépi listázó rutin addig folytatja a listázást, amíg a sor végét jelző 0 értékű bájt után következő bájt értéke is 0 lesz. A mi 2. rutinunk a programsorok végén levő, REM-ben szereplő karakterek ASCII-kódja helyére 0-kat POKE-ol. A listázó rutin ezt fogja sorvégjelnek értelmezni. Ezután következik a tulajdonképpeni ERW (End of RoW) sorvégjelző 0 értékű bájtja, a listázó rutin számára jelezve, hogy vége a programnak. Minden sorszámot ki lehet listázni, de csak a sor végéig. A program futásakor ezeket a bájtokat nem értékeli, mivel REM-ben vannak.

Ha mi magunk szeretnénk kilistázni a programot, például átírás miatt, erre két megoldás is kínálkozik. Az egyszerűbb szerint a 2. programot kell ismét lefuttatni, a 130-as sorban a 0 helyére egy 32 és 159 közötti számot írva. A másik megoldás egy új módszert alkalmaz: programunk első sora elé öt sort be kell írni *(3. program)*:

3. program

```
10 FOR N=sorszám TO sorszám STEP lépésköz
20 POKE 4884, REM (N, 256): POKE 4885, IP (N/256)
30 PRINT 1000
40 NEXT
50 END
```

(Elengedhetetlen, hogy listázandó programunk sorszámai azonos közzel következzenek!)

Ezután adjuk ki a következő parancsot: POKE 4882, 42, majd listázunk. Megjelenik a 3. program öt sora és „főprogramunk" első sora, de a 30-as sorban PRINT helyett LIST utasítás áll. Ha lefuttatjuk a programot, „főprogramunk" sorai rendben listázódnak.

A 20-as sor az N változó értékét — amely jelen esetben csak 1—10000 közötti egész lehet, mivel a LIST utasítás után csak ilyen érték állhat — két bájtra bontja, és a LIST után következő sorszám tárbeli helyére POKE-olja.

Ezzel a módszerrel kiküszöbölhető az IS-BASIC egyik hiányossága is, hogy a RESTORE parancsot nem követheti változónév, csak sorszám, márpedig nagy mennyiségű adat kezelésénél, ha az adatok kiolvasásának helye egy változó értékétől függ, csak SELECT vagy IF utasításokkal operálhatunk, amely sok lehetséges értéknél rendkívül lassú. Viszont ha programunk elejére beírjuk a *4. programot* (a cím keresgélésének elkerülésére írjuk most ezt a program elejére), majd meghívása előtt a sorszámból képzett két bájtot a sorszám helyére rakjuk, a szubrutinhívás után a kiolvasás a képzett sorszámnál kezdődik.

4. program

```
100 DEF REST
110 RESTORE 1000
120 END DEF
.
.
.
500 LET CIM=1000+B
510 POKE 4828, REM (CIM, 256): POKE 4829, IP (CIM/256)
520 CALL REST
530 READ SZAM
.
.
.
```

Az 530-as sorban a kiolvasás az 1000+b-edik sorban kezdődik. Természetesen futáskor a sorszám a 110-es sorban mindig felülíródik az aktuálisan olvasandó sor számával.

A 3. programban látható módon más, eredetileg parancsmódban használható utasítást is beépíthetünk programunkba. Ez különösen kedvező a MERGE utasításnál, ahol változónevet megadva, különböző fájlokat lehet összeolvasni *(5. program)*.

5. program

```
100 GOTO 130
110 PRINT A$
120 GOTO sorszám
130 ! a program többi része
.
.
500 INPUT A$
510 GOTO 110
```

A POKE 4824,45 parancs után a 110-es sorban a PRINT utasítást a MERGE váltja fel, a POKE 4824,67 után pedig a SAVE. (A BASIC tokenek és parancsok táblázata megtalálható a Mikroszámítógép Magazin 1989/4. számában.) A sok ugrabugra azért szükséges, mert DEF blokkot vagy GOSUB-RETURN utasítást használva a 20050 vagy a 10002 számú hibaüzenetet kapjuk.

Dr. Nagy Zoltán

Gaetsch Günterné rajza

SZOFTVER

PATHMINDER

A Westlake Data Corporation cég által forgalmazott PATHMINDER (a továbbiakban: PM) olyan DOS-héj, amely az általános DOS-műveleteket könnyíti meg, valamint szövegszerkesztői feladatokat is el tud látni. A programból közvetlenül futtathatók a .COM, .EXE, .BAT, .BAS kiterjesztésű fájlok. Az alkalmazási (applikációs) menü felhasználásával egyéb alkalmazói programok is egyszerűen futtathatók (maximum 8). A rendszer képes arra, hogy használatáról nyilvántartást vezessen.

Két változatban használható a PC-ben. A virtuális módon működő PMV. EXE fájl a PM.EXE program nem memóriarezidens változata. Mérete 8 kbájt. Ilyenkor a programnak csak egy rezidens része (magja) marad a tárban, és a parancsok aktivizálásakor lép be a lemezen kint lévő megfelelő programrészlet. A program akkor is igen gyors, ha az ismételten betöltődő programváltozatot használjuk. Merevlemezről a betöltés csak egy-két másodpercig tart. A memóriarezidens PM.EXE fájl mérete 80 kbájt. A program színes és monokróm képernyő-adapterrel is működik. A színek beállíthatók, bár az eredmény csak a beállítási parancs végrehajtása után látható.

A PM a belőle hívott, használt programokról nyilvántartást tud vezetni, azaz rögzíti, hogy ki, mikor, milyen programot használt, és mennyi ideig. Ezt tömörített, bináris formában rögzíti, amelyet szövegfájllá átalakítva tudunk értékelni. A LOG2TXT.EXE parancsfájl konvertálja a *.LOG kiterjesztésű fájlt értékelhető szövegfájlra. (A *.LOG fájl magyarázatát lásd később). Megadási módja:

LOG2TX <átalakítandó fájl neve> <átalakított fájl neve>

A PM parancssorba írt, egymástól „/" jellel elválasztott paraméterekkel is indítható. Ezek a következők:

/S: akkor célszerű használni, ha a képernyő a program működése közben „havazik".

/M: ezt a kapcsolót kell használni monokróm képernyő esetén.

/X: az induláskor bejelentkező képernyőt nem jeleníti meg.

/N: beállítja, hogy a program a tárgynapi programhasználatokról ne készítsen összesített kimutatást.

/L: csak az applikációs menü „él".

A .BAS kiterjesztésű BASIC programok futtatásakor a PM először megkérdezi, hogy milyen elérési útvonalon található meg a BASIC-interpreter, majd amíg a rendszerrel dolgozunk, a BASIC programok elindíthatók. Másik lehetőségként a rendszert indító AUTOEXEC.BAT fájlban helyezzük el az elérési útvonalat tartalmazó sort:

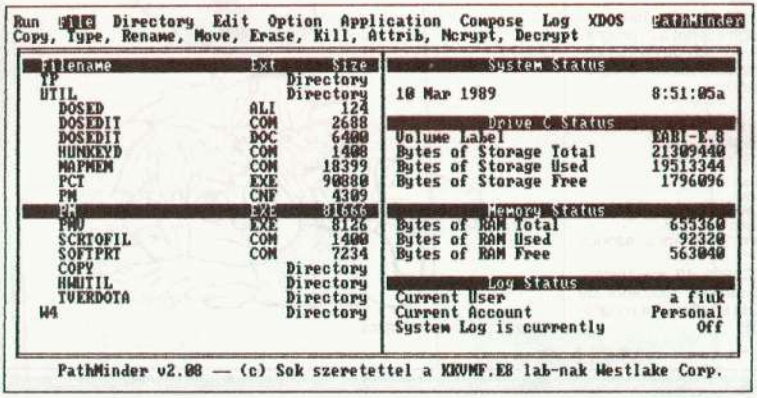
SET BASSPEC=C: \ BASIC \ GWBASIC.EXE

Itt feltételeztük, hogy a GWBASIC-interpreter a C: meghajtó BASIC alkönyvtárában található.

A program indítása után — ha nem tiltjuk le — egy köszönetnyilvánítást tartalmazó szöveg jelenik meg, majd a PM főképernyője.

A képernyő felső részén látható a főmenü. Itt találhatók a programban használható parancsok. A megfelelő parancsot kétféle módon választhatjuk ki: kurzorvezérlők segítségével ráállunk a kívánt parancsra (inverzben jelenik meg), majd az ENTER billentyű leütésével aktivizáljuk. A parancsokat a parancs első betűjének a leütésével is hívhatjuk. Ha az elsőként leírt módon járunk el, akkor a parancs funkciójáról a második sorban rövid tájékoztatást kapunk. Ha a második módon választjuk ki a parancsot, akkor az azonnal végrehajtódik.

A képernyő bal oldalán kapott helyet a lemezen található programok, könyvtá-



rak, fájlok listája. Innen a fájl nevét (File name), kiterjesztését (Ext) és méretét (Size) olvashatjuk le.

A képernyő jobb oldalán a rendszer paraméterei olvashatók. Felül az aktuális dátum és idő, alatta az aktuális meghajtó azonosítója, valamint a meghajtóban lévő lemezről néhány fontos információ: a lemez neve, a lemez kapacitása bájtokban, az elfoglalt lemezterület bájtokban, a szabad lemezterület bájtokban. A felsoroltak alatt a gép belső tárára vonatkozó információk (Memory Status) láthatók:

— a teljes RAM-terület nagysága bájtokban;

— a jelenleg használt RAM-terület nagysága bájtokban;

— a szabad RAM-terület bájtokban.

Legalul a logikai státusz (Log Status) olvasható:

— a jelenlegi felhasználó neve;

— az aktuális azonosító (felhasználóé);

— a nyilvántartás-vezetés be-, illetve kikapcsolt állapotban van-e.

A képernyő bal oldalát a program az F1 billentyű lenyomására Help-képernyővé alakítja. Itt található azon billentyűk használatának magyarázata, amelyekkel a program dolgozik. Az *1. táblázatban* látható ennek a képernyőnek a magyar nyelvű fordítása.

Ha az F4 billentyűt lenyomjuk, megjelenik a kiterjesztett fájlképernyő. Erről leolvasható a fájlok neve (File name), kiterjesztése (Ext), nagysága (Size), a legutóbbi módosításának dátuma (Date), ideje (Time) és attribútuma (File attribute). Itt találkozhatunk a PM egyik hátrányával, nevezetesen, hogy *a rejtett (Hidden) fájlokat nem tudja megjelentetni,* tehát ha rejtett típusú a fájl, akkor nem kerül kijelzésre.

Mivel a PM elsődleges feladata a fájlokkal történő manipuláció, ezért röviden leírjuk, hogyan lehet a fájlokat kijelölni. Egy fájl kijelölése (kiemelés) úgy történik, hogy a fájlt a kurzormozgató billentyűkkel a menüben a más színnel megjelenített kiemelő sorba mozgatjuk (highlighting). Egy vagy több fájlt kijelölhetünk úgy, hogy a kiemelő sorba mozgatjuk, majd a SPACE billentyűvel egy kis téglalap alakú jelet helyezünk el a fájl sorában (mark). A kijelölés a SPACE ismételt megnyomásával megszüntethető. Ha a kijelöléskor egy könyvtár áll a kiemelő sorban, akkor a könyvtárban lévő összes fájl kijelölésre kerül.

Billentyű	Magyarázat
Enter	Az inverzben lévő parancs végrehajtása
<- (Crsl)	Előző parancs kijelölése
-> (Crsr)	Következő parancs kijelölése
^ (Crsu)	Előző fájl kijelölése
v (Crsd)	Következő fájl kijelölése
PgUp	Fájl lista felfelé történő lapozása
PgDn	Fájl lista lefelé történő lapozása
Home	Fájl lista elejére
End	Fájl lista végére
Grey -	(Szürke "-" gomb) Előző könyvtár kijelölése
Grey +	(Szürke "+" gomb) Következő könyvtár kijelölése
Ins	A kijelölt könyvtár megnyitása
Del	Az aktuális könyvtár zárása
Esc	Előző menüpontra való visszatérés
F1	Help képernyő
F3	Az a státuszképernyő, amit bejelentkezés után látunk
F4	Kiterjesztett fájlképernyő

1. táblázat

A főmenü parancsai

RUN. Feladata a kiemelt program futtatása. Ha nem futtatható program van kiválasztva (például egy alkönyvtár neve), akkor hibaüzenetet kapunk: „csak .bat, .com, .exe és .bas kiterjesztésű fájlok futtathatók". Ha a RUN-nal elindított program lefutott, a „Strike a key when ready" üzenetet kapjuk, azaz üssünk le egy billentyűt, ha folytatni akarjuk a munkát a PM-mel. Mivel a PM rezidens, ezután visszatérünk a főmenübe.

FILE. Innen fájlokkal végezhető műveleteket jelölhetünk ki egy újabb almenüből. Mivel itt fájlokkal végezhető műveletek vannak, a műveletekben részt vevő fájlt vagy fájlokat ki kell jelölni. Az almenüből kiválasztunk egy parancsot, majd az ENTER leütésével aktivizáljuk. A FILE parancs almenüjében lévő parancsok:

COPY. Az azonos nevű DOS-parancscsal egyezik meg. Segítségével a kijelölt fájlok másolhatók egyik alkönyvtárból a másikba, illetve egyik lemezről a másikra. A COPY parancs két választási lehetőséget biztosít: alkönyvtár (Subdirectory) vagy meghajtó (Drive). Az alkönyvtár választásakor alkönyvtárak közötti másolás történik, Drive választásakor pedig lemezek közötti másolás lehetséges. Az opciók kiválasztása után lehet a fájlokat a fent említett módon kijelölni. Ha a Subdirectoryt választottuk, akkor a fájlok kijelölése után ENTER-t ütünk, utána rá kell állni arra az alkönyvtárra, ahová be szeretnénk másolni a fájlokat, majd ismét ENTER-t adunk. Ekkor megtörténik a másolás. Drive-választáskor a rendszer megkérdezi, hogy melyik meghajtóban van a céllemez, majd megtörténik a másolás. Másoláskor az azonos nevű fájlok felülíródnak. Ha a forráslemez nem a winchester, akkor az eljárás a következő: először a főmenü OPTION, majd DRIVE parancsai útján választhatjuk ki a kívánt meghajtót. Figyelem: *Egy meghajtó esetén nem lehet a lemezt cseréléssel másolni!* Ezután adható ki a FILE, majd a COPY parancs. Ettől kezdve a már leírtak szerint történik a másolás. A másolás befejezésekor a program a forráslemezt tartalmazó meghajtóra áll rá.

TYPE. Azonos a DOS-parancscsal: képernyőre listázza a kiválasztott fájl tartalmát.

RENAME. Azonos a DOS-parancscsal: a kiválasztott fájlt átnevezi. A név mellett a fájlkiterjesztést is meg kell adni.

MOVE. Ez a parancs a DOS-ban nem létezik. A parancs áthelyezi a kiválasztott fájlt egyik alkönyvtárból egy másik alkönyvtárba. Magától értetődően ezt a parancsot csak egy lemezen belül használhatjuk. A parancs eredménye, hogy a kiválasztott fájl elérési útvonala megváltozik.

ERASE. A DOS DELETE parancsával egyezik meg, tehát a kiválasztott fájlt logikailag törli. Ez azt jelenti, hogy a lemezkönyvtárban a fájl nevének első betűjét „@" karakterre cseréli ki, azaz felszabadítja a fájl helyét a lemezen. Ha ilyen módon töröltünk egy fájlt, és még nem történt írás a törölt fájl helyére, akkor ezt a fájlt vissza lehet állítani. *A PATHMINDER erre a visszaállításra nem képes.*

KILL. Ez a parancs sem létezik a DOS-ban. Az ERASE paranccsal ellentétben a KILL parancs fizikai törlést hajt végre. Először az állomány tartalmát nullázza, csak ezután szabadítja fel az állomány helyét a lemezen. Az ily módon törölt állományok visszaállítására nincs mód.

ATTRIB. E parancs segítségével a fájlattribútumok át-, illetve beállíthatók. Mivel a PM hidden fájlokat nem tud kijelezni, csak a read only és az archív attribútumok állíthatók be.

NCRYPT, DECRYPT. A PM-nek ez a két szolgáltatása egyedi. Az NCRYPT választásakor a kijelölt fájl kódolható, illetéktelen használat ellen védhető. Ehhez a kódoláshoz egy max. 80 karakterből álló jelszót kell megadnunk. A kódoláshoz egy rejtő eljárás szerint a jelszóból és a fájlból egy, más által hozzáférhetetlen állományt hoz létre a program. A dekódolást a DECRYPT végzi. A dekódolás az általunk megadott jelszóra megy végbe, tehát mi is csak a DECRYPT-tel férünk hozzá a kódolt fájlhoz. A kódolt fájlok kiterjesztése .000-tól .999-ig terjedhet. Például:

Kódolandó fájlnév		*Kódolt fájlnév*
MINTA.DOC	→	MINTA .000
PÉLDA.TXT	→	PÉLDA .000
MINTA.HLP	→	MINTA .001

DIRECTORY. Könyvtárakkal kapcsolatos műveletek jelölhetők ki egy újabb almenüből. Az almenü parancsai:

OPEN. Alkönyvtár módosítása. A kiválasztott alkönyvtárban található állományok válnak láthatóvá a parancs hatására. A megjelenítés úgy történik, hogy az alkönyvtár alatti állományok vagy alkönyvtárak két karakterrel jobbra tolódnak, így téve szemléletessé hovatartozásukat.

CLOSE. Alkönyvtár zárása. A kiválasztott alkönyvtár alatti állományok kijelzése megszűnik. Megjegyzés: ezzel a két paranccsal egyenértékű az „Ins", illetve a „Del" billentyű hatása, azzal a különbséggel, hogy nem kell használatukhoz a DIRECTORY parancsra állni.

SORT. Rendezhetjük az alkönyvtár alatti fájlokat növekvő sorrendbe, különböző rendezési szempontok szerint. Megadható rendezési opciók: név, kiterjesztés, dátum, idő. *A rendezés eredményét nem lehet a könyvtárban ténylegesen rögzíteni.*

MAKE. Megegyezik a DOS MD parancsával. A kijelölt könyvtár alatt új könyvtárat hoz létre.

REMOVE. Megegyezik a DOS RD parancsával. A kijelölt könyvtárat lehet törölni, de csak abban az esetben, ha az már nem tartalmaz fájlokat.

_ Kurzormozgatás a szerkesztett szövegben

karakterenként jobbra	->
karakterenként balra	<-
szavanként jobbra	Ctrl és ->
szavanként balra	Ctrl és <-
sor elejére	Home
sor végére	End
képernyő elejére	Ctrl + Home
képernyő végére	Ctrl + End
előző lap	Pg Up
következő lap	Pg Dn
fájl elejére	Ctrl + Pg Up
fájl végére	Ctrl + Pg Dn

Megjegyzés: szóhatár: a szóköz és az összes írásjel

Beszurás és törlés

Insert mód be/ki: Ins
Előző karakter törlése: Bksp
Karakter törlés: Del
Sor törlés: Ctrl + Bksp

Megjegyzés: sor törléskor a törölt sor a kapcsos zárójelek közé kerül.

2. táblázat

EDIT. Ezzel a paranccsal aktivizálható a PM-be épített szövegszerkesztő. Ha új szöveges fájlt akarunk létrehozni, akkor a parancs kiadása előtt arra az alkönyvtárra kell állni, ahová a fájlt tenni akarjuk. Ha már meglévő fájlt szeretnénk editálni, akkor ráállunk a fájlra. Ezek után aktivizálhatjuk az EDIT parancsot, majd a program betölti a fájlt.

A képernyő legfelső sorában balról jobbra haladva látható a szerkesztett fájl neve (új fájl esetén itt kell megadni a fájl nevét). Itt szeretnénk felhívni a figyelmet arra, hogy a kiterjesztéshez a fájlnév után a pontot is be kell adni, mert különben a fájlnak nem lesz kiterjesztése. Ezután a kurzor aktuális sorának (Line) és oszlopának (Column) számjegyes kijelzése következik. A kurzor ezen a pozíción található a szövegben. A „Position" a kurzor abszolút helyét adja meg a szövegben, azaz azt, hogy a szöveg elejétől hányadik karakterpozíció. Például ha a második sor első karakterén áll a kurzor, akkor a position értéke 81 lesz. Utolsóként a szövegbeírási mód kerül kijelzésre, ami lehet beszúrás (insert) vagy felülírás (overtype). A második sorban kapcsos zárójeleket láthatunk, ahová a szövegből kiemelt szövegrészek kerülnek. A kiemelt szövegrészből csak az első 80 karakter jelenik meg, de ez nem akadálya annak, hogy ennél hosszabb szöveget is kiemelhessünk. Ezután következik a szerkesztendő szöveg. A legalsó sorban az F.. billentyűkhöz rendelt parancsok láthatók.

A funkcióbillentyűk

F1 — Help. Hatására a szerkesztett szöveg eltűnik, és a szövegszerkesztő által használt speciális billentyűk, valamint a hozzárendelt parancsok leírása jelenik meg.

F2 — Format. Szöveg tördelése, az előzőleg beállított margó szerint.

F3 — Delete. Törlés csak a szövegrész kijelölése után adható ki. Törléskor a törölt szöveg a kapcsos zárójelek közé kerül, így szövegmásolás vagy -mozgatás parancsra a szövegbe bárhová beszúrhatjuk, visszahívhatjuk.

F4 — Insert. Beszúrás. Kiemelt szövegrész beszúrása a szerkesztett szövegbe. A beszúrás a kurzor pozíciójától kezdődik.

F5 — Copy. Ez a parancs a következőképpen működik. F6 megnyomása után a kurzorvezérlővel a kiemelendő szöveg végére állunk, majd F5-öt megnyomva bekerül a szöveg a kapcsos zárójelek közé.

F6 — Mark. Kiemelendő szöveg kezdetének kijelölése.

F7 — Replace. Keresés, helyettesítés. A Search opció alatt megadott karakterláncot cseréli ki az általunk megadott karakterláncra.

F8 — Find. Keresés. A Search opció alatt megadott karakterlánc keresésének elindítása. Megjegyzés: a Replace és a Find a keresést a kurzor pozíciójától jobbra és lefelé kezdi.

F9 — Update. Aktualizálás. Szerkesztés közbeni mentésre szolgál. Tehát az F9 leütéséig szerkesztett szöveget menti el úgy, hogy nem kell kilépni a parancsmódba.

F10 — Esc. Escape karakteres megjelenítése a fájlban. Mivel az Alt-27 billentyűkombinációval nem váltható ki a szövegfájlban az Esc karakteres képe, ezért az F10 billentyűhöz hozzárendelték az Esc képét. Erre azért van szükség, mert általában az IBM PC családhoz kapcsolható printerek vezérlőkódjai Esc-pel kezdődnek.

A szövegszerkesztő parancsmenüje

Ebbe a menürészbe az Esc billentyű lenyomásával kerülhetünk. Ekkor a képernyő felső sorában megjelennek a kiadható parancsok. A parancs kiválasztása, aktivizálása a szokásos módon történik.

DISK. Megadásával egy újabb almenü pontjaiból választhatunk.

Save: az egész fájl mentése.

Update: szövegszerkesztés közbeni mentés (lásd F9 billentyű)

Put: kijelölt szöveg írása egy adott fájlba vagy a printerre.

Get: szövegfájl hívása lemezről. Itt lehetőség van a szerkesztett szövegbe külső fájl beszúrására (insert) vagy a szerkesztett fájl egy részének felülírására (overtype). Az általunk megadott fájlt Insert módban beírja a kurzor elé, vagy Overtype módban a kurzor utáni részt felülírja a megadott fájllal.

FORMAT

Center: a szerkesztett sort középre helyezi.

Left: a sort a bal margóhoz igazítja.

Right: a sort a jobb margóhoz igazítja.

Format: két üres sor közötti szövegből kiveszi az újsor-karaktereket, és teljes sorokat képez a margón belül.

BLOCK. Kijelölt blokkal végzett műveletek jelölhetők ki. Beszúrás, törlés, másolás parancsokat adhatunk ki egy almenüből. Ehhez magától értetődően ki kell jelölni a műveletben részt vevő szövegrészletet.

Insert: a kijelölt szövegrészt beszúrja a kurzor mögé.

Delete: a kijelölt szövegrészt törli, de az nem vész el, az *Inserttel* visszahívható. Így lehet szövegrészeket egyik helyről a másikra áthelyezni.

Copy: a kijelölt szöveget elteszi, amely az *Inserttel* visszahívható. Így lehet szövegrészeket megduplázni.

MARK. Szövegrészek kijelölése (lásd F6). A kijelölés kezdetekor kell lenyomni, majd a kurzorral a kívánt szövegrész végére állni. Ismételt alkalmazásával a kijelölés megszüntethető.

JUMP. Itt megadható a szövegnek az a része, ahová a kurzort szeretnénk állítani.

Begin: lépés a fájl elejére.

End: lépés a fájl végére.

First: a képernyő első sora.

Last: a képernyő utolsó sora.

SEARCH. Itt adhatjuk meg azt a karaktersorozatot, amit keresünk, valamint azt a karaktersorozatot, amit ki akarunk cserélni (F8, F7).

Find: a kurzor utáni szövegrészben megkeresi a beírt karaktersorozat első előfordulási helyét (F8).

Replace: a beírt karaktersorozatot kicseréli a kívántra (F7).

OPTION

Update: ez az opció állítja be azt, hogy a Save parancs felülírja-e a szerkesztett fájlt a lemezen, vagy az előző változatot .BAK kiterjesztésűre cserélje ki. Ez utóbbi az alapértelmezés.

Tab set: tabulátorok állítása (max. 10 darab).

Auto Indent: az előző sorban elkezdett bekezdéssel kezdi az új sort.

Position: ha igennel (yes) válaszolunk, akkor a szöveg szerkesztésekor a felső sorban látható lesz a szerkesztett fájl neve, a kurzor pozíciója. Ha nemmel (no) válaszolunk, akkor a felső sor üres lesz.

Margin: itt állíthatjuk be a margót, vagyis az egy sorban elhelyezkedő karakterek számát. Alapértelmezésben ez az érték 80.

Wrap: yes esetén a margón túlnyúló szót automatikusan átteszi az új sorba.

Status: az előzőek beállítását mutatja.

HELP. Az editor parancskészletét jeleníti meg, amit a *2. táblázatban* foglaltunk össze.

OPTION. A főmenü e parancsának kiadásakor egy almenübe jutunk. Az almenü opciói:

DRIVE: Az általunk használni kívánt aktív lemezegység kiválasztása itt történik (lásd COPY parancs). Ha munka közben lemezt cseréltünk, akkor újra ki kell választani ugyanazt a meghajtót, azért, hogy az új lemez tartalomjegyzékét beolvassa a program.

VIEW. Ugyanaz a szerepe, mint az F1, F3, F4 billentyűknek a főmenüben.

BEEP. Hiba esetére a hangjelzés be-, illetve kikapcsolása itt történhet.

ATTRIB. Színes monitor esetén a képernyő színei innen állíthatók be. Monokróm képernyő használatakor a kiírások intenzitása is itt állítható.

USER. Itt állítható be, hogy a programot használó mennyire járatos a PATHMINDER használatában. *Novice-t* megadva kezdő, *Expertet* megadva gyakorlott felhasználót tételez fel a program, és egyes műveleteket (például másolás) külön megerősítés nélkül hajt végre. Ezzel az opcióval az egyes parancsokhoz írt magyarázó szövegek részletessége is beállítható.

EDIT. Kiválaszthatjuk, hogy a szövegszerkesztő fél vagy egész képernyős legyen-e. Fél képernyő esetén a szerkesztendő szöveg a képernyő jobb oldalán lesz látható, a képernyő bal oldalán pedig a fájllista jelenik meg.

MENU. Ebben az opcióban adhatók meg az alkalmazói (applikációs) programok futtatásához szükséges paraméterek.

Select: az applikációs program kiválasztása.

Edit: az applikációs program szerkesztése. Itt kell megadni az applikáció nevét (Name), leírását (Description), az útvonalat (Path), a fájl nevét, kiterjesztését (File Spec.), a fájl parancssorának jellemzőit (Command line) (lásd még: **APPLICATION**).

Help Edit: az applikációs program helpképernyőjének szerkesztése.

Remove: az applikáció törlése a menüből.

Quit: kilépés.

STATUS. Lekérdezhetők a PM legfontosabb beállított paraméterei: meghajtó, rendezési szempont, hangjelzés állapota, milyen szerkesztési mód az aktív.

WRITE: A beállított paramétereket kimenti a PM.CNF fájlba (színek, szövegszerkesztő, applikációs programok paraméterei stb.). A PM későbbi használatakor a program innen olvassa be a paramétereit, tehát amit most beállítottunk, az későbbi indításkor már alapértelmezés lesz.

APPLICATION. Amint azt már az ismertetés elején leírtuk, a RUN parancs kiadásával az .EXE, .COM, .BAT és a .BAS kiterjesztésű programok a PM-ből elindíthatók, azaz futtathatók. Milyen jó lenne, ha például a Turbo Pascalt úgy tudnánk használni, hogy a PM-ben ráállnánk a betöltendő fájlra, és aktivizálnánk a Turbót, aminek hatására a program a kijelölt fájllal betöltődne. Nos, ez a parancs ezt teszi lehetővé. Maximum nyolc ilyen alkalmazói program hívása oldható meg. Azok a programok hívhatók azonosítójuk kezdőbetűjének leütésével, amelyeket az OPTION MENU-jében megadtunk.

COMPOSE. Közvetlen DOS-parancsot adhatunk ki úgy, mintha közvetlenül az operációs rendszerben lennénk. A főmenübe az Esc-pel térhetünk vissza.

```
RUN   FILE   DIRECTORY  EDIT      OPTION     APPL.   COMPOSE   LOG   XDOS

Kiterjesztes:  COPY          OPEN      DISK       DRIVE      A       DOS parancs   YES   YES
   .EXE          SUBDIRECTORY  CLOSE     SAVE       VIEW       B       kiadasa       NO    NO
   .COM          DRIVE         SORT      UPDATE(F9)   STATUS     ..                    ACCOUNT
   .BAT       TYPE          MAKE      PUT          DATE/TIME  ..                    USER
   .BAS       RENAME        REMOVE       FILE       HELP       (MAX. 8 DB.)          QUIT
              MOVE                         PRINTER  BEEP
              ERASE                     GET        ATTRIB
              KILL                   FORMAT          FOREGROUND
              ATTRIB                    CENTER       BACKGROUND
                ARCHIVE                 LEFT       USER MODE
                UNARCHIVE               RIGHT         NOVICE
                PROTECT                 FORMAT(F2)    EXPERT
                DEPROTECT             BLOCK        EDIT MODE
              NCRYPT                    INSERT(F4)    FULL SCREEN
              DECRYPY                   DELETE(F3)    DATA WINDOW
                                        COPY(F5)   PAUSE
                                     MARK(F6)      MENU(APPLICATION)
                                     JUMP             SELECT
                                        BEGIN         EDIT
                                        END           HELP EDIT
                                        FIRST         REMOVE
                                        LAST          QUIT
                                     SEARCH        STATUS
                                        FIND(F8)   WRITE
                                        REPLACE(F7)
                                     OPTION
                                        UPDATE
                                        TAB
                                        AUTO-INDENT
                                        POS
                                        MARGIN
                                        WRAP
                                        STATUS
                                     HELP(EDITOR)
                                     QUIT

F1:  MENU HELP
F3:  DRIVE/MEMORY STATUS KIJELZES
F4:  FILE DATUM/IDO KIJELZES
```

A PATHMINDER menürendszere

LOG. Nyilvántartáshoz szükséges parancsok, opciók kiadása, beállítása egy almenün keresztül érhető el.

A LOG parancs opciói

YES. Ha ezt választjuk, akkor a program készít nyilvántartást a PM-ből hívott programokról. Rögzíti a hívott, szerkesztett programok nevét, ki használta azt a programot és mikor. Ezeket az adatokat .LOG kiterjesztésű fájlban rögzíti. Vigyázat, ha lemezt cserélünk, és nincs a lemezen PM.LOG fájl, a program létre fog hozni egyet! Ez az opció a Log stat képernyőrészen úgy jelenik meg, hogy a „System Log is currently" mellett „on" látható.

NO. Ebben az esetben nem készít nyilvántartást a PATHMINDER. „System Log . . . " mellett „off" látható.

ACCOUNT. Itt a felhasználó azonosítóját adhatjuk meg, ami lehet „personal" vagy 1—9-ig terjedő szám. A log stat képernyőn a „Current Account" mellett látható a beállított azonosító.

USER. A programmal dolgozó felhasználó neve itt adható meg. Log stat képernyőrészen a „Current User" mellett jelenik meg ez a név.

XDOS. Ezzel a paranccsal léphetünk ki a PM-ből, miután a „Jól meggondolta?" kérdésre „Y" választ adtunk.

A menük közötti eligazodást segítendő, az *ábrán* látható a PATHMINDER menürendszerének felépítése.

Mindent egybevetve, a PM nagyon jól használható program, valóban igen kellemes felhasználói környezetet biztosít. Kár, hogy a 2.08-as verziót a cég nem fejlesztette tovább. Merevlemezzel ellátott gépeken a PMV változatot célszerű használni, mert így nagyon kevés memóriát igényel. Hiányosságaira már a fentiekben is utaltunk. Még néhány, ami inkább a továbbfejlesztés elmaradásából adódik:

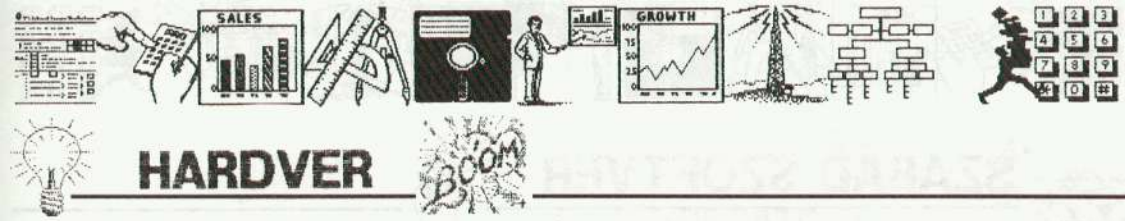
— az editor maximum 64 kbájtos szövegfájlokat tud kezelni, sort kiegyenlíteni nem tud,

— a nagy felbontású képernyőket (EGA, VGA) nem tudja kezelni.

Mindezek ellenére a program megvételét és használatát feltétlenül javasoljuk.

FIGYELEM!

A PÉCÉZZÜNK rovatban megjelent cikkek szövege szövegfájlok formájában, valamint az „Ajándék" szabad szoftver 360 kbájtos DS-DD lemezen, utánvéttel, önköltségi (lemezár, lemezmásolás, postázás) 300 forintos áron megrendelhető. Cím: Koncz Edit, Budapest, Kunigunda u. 44. 1037



PC-vizsgáló áramkör

Egy számítógép működésképtelenségének felderítése nem egyszerű feladat. A hiba megállapítását segíti, ha megvizsgáljuk a rendszer fontosabb jeleinek állapotát. Az ábrán látható áramkör ezt a feladatot látja el, a legtöbb PC-ben nem használt 8087-es matematikai koprocesszor IC-foglalatának segítségével. Az áramkört úgy alakítjuk ki, hogy az a koprocesszor-foglalatba bedugható, csatlakoztatható legyen. Az áramkört egy megfelelő, 40 lábú IC-foglalatra építjük fel.

Természetesen az áramkör fixen is beépíthető a gépbe, így mindig láthatjuk rendszerünk állapotát.

A 8288-as buszvezérlő a processzor S0, S1, S2 státuszjeleinek a felhasználásával visszaállítja a vezérlőjeleket.

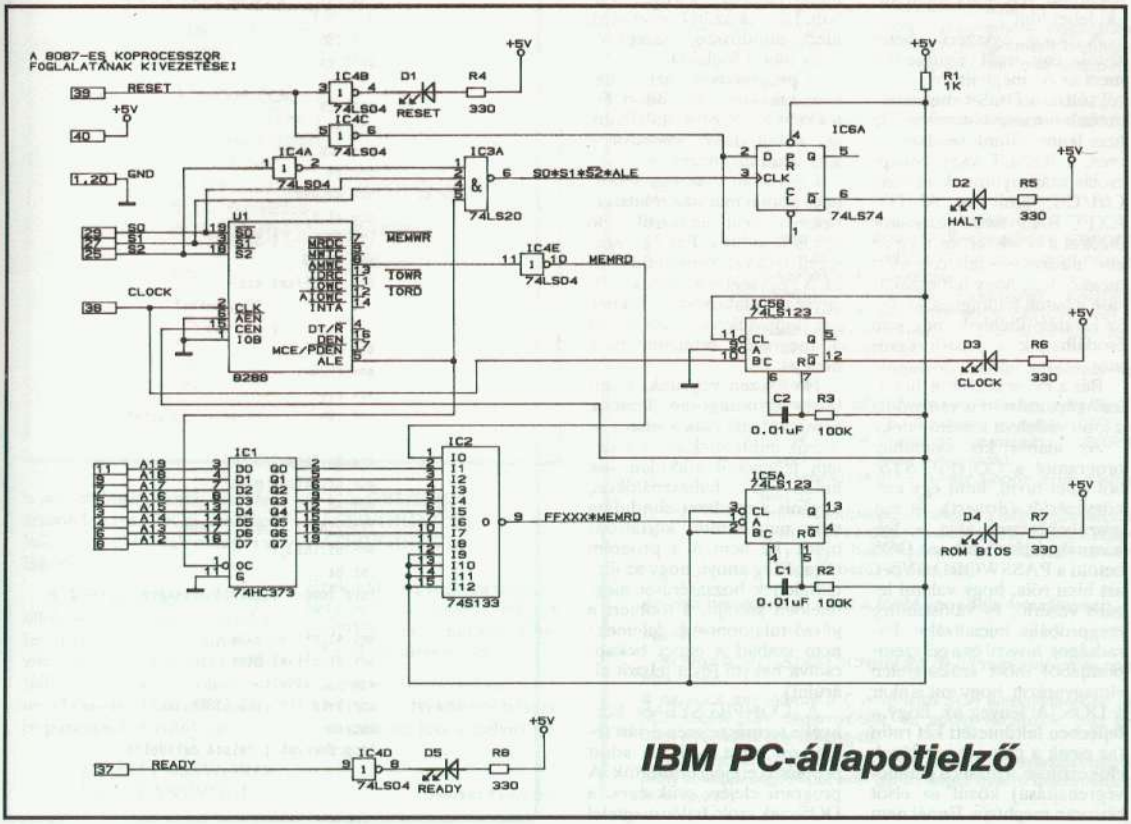
A RESET LED akkor világít, ha a processzor RESET jele aktív.

A CLOCK LED jelzi az órajel meglétét.

A READY LED jelzi a READY állapotot.

A ROM BIOS LED világít, mikor a rendszer RESET után a ROM-BIOS-ban lévő programot hajtja végre.

A HALT LED jelzi, mikor a ROM-BIOS közvetlenül a bekapcsolás után lefutó öntesztelő programja olyan súlyos hibát talál, hogy a processzort HALT állapotba vezérli.



IBM PC-állapotjelző

SZABAD SZOFTVER

Egy másik kulcs . . .

A Magazin 1989/8. számának 6. oldalán megjelent egy program, amely az illegális hozzáférést volt hivatott meggátolni egy jelszó bekérésével. A programot az AUTOEXEC.BAT aktivizálta, ami egyben az egyetlen gondot is jelentette vele kapcsolatban: „ki lehet lőni"!

A dolog egyszerű: beteszünk egy saját rendszerlemezt az A: meghajtóba, és arról töltve a DOS-t, hozzáférhetünk a merevlemezhez. Ha nem lenne nálunk rendszerlemez, a RESET vagy bekapcsolás után nyomunk néhány Ctrl/C-t, amit az AUTOEXEC.BAT feldolgozásakor érzékel a rendszer, és még az első utasítás — a jelszó — előtt megkérdezi, hogy félbeszakítsa-e a batch feldolgozását. Így az elvetemültebbek meg sem csodálhatják a jelszóprogramot.

Bár a PC sem „feltörthetetlen" gép, azért itt is van módja a jobb védelem készítésének.

Az alábbi kis assembler programot a CONFIG.SYS-ből lehet hívni, mint egy eszközvezérlőt (drivert), és egy jelszóbekéréssel állja a felhasználó útját. Amikor a DOS betölti a PASSWORD.SYS-t, azt hiszi róla, hogy valami legális vezérlő, és valószínűleg megpróbálja inicializálni. Fáradságos, hosszú és a cél szempontjából most szükségtelen elmagyarázni, hogy mit is akar a DOS. A lényeg az, hogy a fejlécben feltüntetett két rutin (az egyik a parancs vételének előkészítése, a másik a parancs végrehajtása) közül az elsőt biztosan meghívja. Ennél nem is kell több. Ide került a jelszóbekérés. A másik rutin egy sima RET utasítás. Az első futás után többször nem kerül a programra a vezérlés, mivel mi nem hivatkozunk rá.

A megoldás egyetlen hátránya, hogy programunk bent marad ballasztnak a memóriában. Ez csak azért bocsánatos, mert mindössze százegynéhány bájtot foglal el.

A programban a jelszó három karaktere egy adott értékkel XOR-olva található, így a fájl direkt olvasásával sem lehet megnézni.

Ezek után csak egy sebezhető pontja marad a rendszernek: a saját lemezről való rendszerindítás. Ezt egy egyszerű trükkel kerülhetjük ki: az A: egységbe átdugjuk a B: egység csatlakozóját. Akiknek két floppyjuk van, azoknál ez a megoldás értelemszerűen nem jó.

Ha készen vagyunk, a dobozt biztonságosan lezárva, gépünket már csak a jelszót ismerők indíthatják el. A védelem jellegéből adódóan sok múlik a felhasználókon, ugyanis a rendszer elindulása után már semmi korlátozás nincs. Ez nem is a program célja. Elég annyi, hogy az illetéktelenek hozzáférését megfelelően gátolja, a többiért a jelszó-tulajdonosok felelnek: nem szabad a gépet bekapcsolva hagyni (és a jelszót elárulni).

A CONFIG.SYS és környéke természetesen óriási téma, mi most csak az adott probléma erejéig érintettük. A program elejére szükséges, a DOS-nak szóló fejléc megfelel

```
PasswordS  segment 'code'
           Assume cs:PasswordS,ds:PasswordS
Passwlen   equ 3  ; Jelszóhossz
EXORMask   equ 33 ; Titkosítómaszk

           dw 0ffffh          ; Fejléc a DOS-nak
           dw 00000h
           dw 00000h
           dw offset Input
           dw offset Process
           db 'PASSWORD'

Input      proc far

           mov ah,3   ; Kurzorméret mentése
           mov bh,0
           int 10h
           push cx

           push cs
           pop ds
           push cs
           pop es
again:     mov ah,0eh
           mov al,07
           int 10h     ; Sípolás
           mov ah,09
           mov dx,offset szov
           int 21h     ; "Password:"
                       ; kiírása
           mov ah,01
           mov ch,0
           mov cl,7
           int 10h     ; saját kurzorméret

           mov cx,Passwlen
           mov bx,offset pass
loop:      mov ah,07
           int 21h     ; kar. input
           mov ds:[bx],al
           inc bx
           loop loop   ; jelszó bekérése

           mov si,offset pasw
           mov di,offset pass
           mov cx,Passwlen
decrypt:   xor byte ptr [si],EXORMask
           inc si
           loop decrypt ; jelszó dekódolás
```

```
            mov si,offset pasw
            mov cx,Passwlen

            repe cmpsb  ; összehasonlítás
            jne again   ; ha nem jó, újra

            pop cx      ; Kurzorméret visszállítása
            mov ah,1
            int 10h

            mov dx,offset wond
            mov ah,9
            int 21h     ; "Accepted"
                                ; kiiratása

            ret

            ; A betuk EXORMaskkal EXOR-olva
            ; Ide irhatjuk be a saját jelszavunk
            ; betűit

pasw:       db 'D' xor EXORMask
            db 'R' xor EXORMask
            db 'K' xor EXORMask

pass:       db '000'    ; puffer a bekérendő jelszónak
szov:       db 13,'Password:$'
wond:       db 13,'Accepted  ',13,10,10,'$'

Input       endp

Process     proc far
            ret
Process     endp
PasswordS   ends
            end
```

a követelményeknek, részletesebb leírása a DOS-szal foglalkozó szakirodalomban található.

A közölt programot bárki elképzelése szerint továbbfejlesztheti. A szöveget elhelyezzük a PASSWORD.ASM fájlban, majd a makroassemblerrel való fordításhoz az alábbi parancsokat adjuk ki:

C > MASM PASSWORD;
C > LINK PASSWORD;
C > EXE2BIN PASSWORD.EXE PASSWORD.SYS

Ezek után a program üzembe helyezése a következő. A CONFIG.SYS fájlban elhelyezzük az alábbi sort:

device=password.sys

Az újabb rendszertöltéskor már szükség lesz a helyes jelszóra.

Szinnyei Gerzson

AJÁNDÉK

SCRBATCH: képernyőszerkesztő és batch programozást segítő program

Ajándék programjaink sorát megint egy jól használható programmal folytatjuk, az 1985-ben, Frank M. Ramaekers jr. által írt, eredeti nevén The Screen Image Editor and Batch Utility szoftverrel.

A program két részből áll. Az első a SCREDIT program, amelylyel egy megjeleníthető képernyőt szerkeszthetünk meg, és azt egy fájlba írhatjuk. A második a SCRBATCH program, amellyel a megszerkesztett képernyőt megjelenítve, könnyen tudunk programokat a billentyűzet felhasználásával batch programból indítani.

A SCREDIT a Screen Image Editor (=képernyőszerkesztő) lehetővé teszi, hogy a képernyőn tetszőleges szöveget, menüket, segítő képernyőket, illetve karakteres grafikát megszerkesszünk az IBM teljes karakterkészletének és a képernyő-attribútumoknak a felhasználásával. A 80H-FFH tartományba eső grafikus karakterek is a billentyűzeten keresztül írhatók a képernyőre, mivel a grafikus karakterkészlethez is egy billentyűzetkiosztás tartozik, és ez átváltható (normál/grafikus).

A SCRBATCH program a SCREDIT programmal megszerkesztett képernyő felhasználásával batch programból hívható, és menüszerűen kialakított batch programozást tesz lehetővé.

A parancs alakja:

SCRBATCH d: útvonal fájlnév.ext /a /c /i /n /o /pnnn /q

Itt az első paraméter a megjelenítendő képet tartalmazó fájlt specifikálja.

d:	*— meghajtó.*
útvonal	*— a fájl elérési útja.*
fájlnév	*— a fájl neve.*
.ext	*— a kiterjesztés (.SCR a feltételezett).*
/a	*— billentyűre kérdezés (visszatérés: ERRORLEVEL=ASCII, ahol ASCII a megnyomott billentyű nagybetűs ASCII-kódja).*
/c	*— billentyűre kérdezés (visszatérés: ERRORLEVEL=0, ha A betűt nyomtunk, ERRORLEVEL=1, ha B betűt nyomtunk, . . . ERRORLEVEL=25, ha Z betűt nyomtunk).*
/i	*— a 25. sorban megjelenő prompt jelsorozat megadása.*
/n	*— numerikus billentyűre kérdezés (visszatérés: ERRORLEVEL=0, ha 0 számot nyomtunk, ERRORLEVEL=1, ha 1 számot nyomtunk, . . . ERRORLEVEL=9, ha 9-est nyomtunk).*
/o	*— a 25. sor színe (07 — fekete alapon fehér a feltételezett).*
/pnnn	*— nnn másodpercig várakozik a billentyűnyomásra.*
/q	*— igen/nem kérdés (visszatérés: ERRORLEVEL=0, ha y vagy Y betűt nyomtunk, ERRORLEVEL=1, ha n vagy N betűt nyomtunk).*

Az első paraméteren kívül a többi sorrendje tetszőleges.

Példa:

SCRBATCH C:\SDCR\SCRHELPA /P /I'Press any key to continue' /087

A parancs megjeleníti a C meghajtó SCR alkönyvtárában lévő SCRHELPA.SCR képernyőfájlt, és addig várakozik, míg meg nem nyomunk egy billentyűt.

A fent ismertetett két programon kívül a szoftverhez még néhány további segítséget adó fájl és angol nyelvű dokumentáció is tartozik.

A versenyképesség építőkövei

Sokan kérdőjelezték meg a gombamód elszaporodott kisvállalkozások között a hagyományos, nagy számítástechnikai szervezetek helyét és szerepét. Sokan mármár temetni kezdték a „nagyokat", mondván, rugalmatlan struktúrájuknál fogva képtelenek alkalmazkodni a megváltozott piaci viszonyokhoz, a gyorsabban reagálni tudó „kicsik" egyre újabb és újabb területeket halásznak el előlük. A Kereskedelmi Szervezési Intézetet többnyire még mindig úgy tartják számon, mint egy nagyszámítógépes bérmunkairodát, ahová viszik meg ahonnan hozzák a bizonylatokat, leporellókötegeket. Pedig a valóság ma már egészen más: akárhol forduljon is meg az ember a főváros kereskedelmileg exponált vállalatainál, lépten-nyomon találkozhat a KERSZI helyszíni szolgáltatásaival. Arról már nem is beszélve, hogy a „nagy" KERSZI számos leányvállalatot és korlátolt felelősségű társaságot is magáénak tudhat, általuk ötvözve a kétféle struktúrából fakadó előnyöket.

A Kereskedelmi Szervezési Intézet két évtizedes múltra tekinthet vissza a számítástechnikai szakmában, ez a rutin teszi lehetővé, hogy eredményesen dolgozza ki és üzemeltesse azokat a számítástechnikai és alkalmazási célrendszereket, amelyeket a kereskedelmi, a vendéglátóipari és idegenforgalmi szervezési tevékenységek igényelnek.

A referenciák azt igazolják, hogy az Intézet az igények felmérésétől a megvalósításig a lehetséges optimális megoldást kínálja: segítséget nyújt a rentábilitási számításokhoz, az áruforgalom megszervezéséhez, a szállodai helyfoglalási rendszerek kialakításához, valamint a raktárirányítás, a készletgazdálkodás, az áruszállítás, a kereskedelmi értékesítés, a korszerű vonalkód-alkalmazás megszervezéséhez. Foglalkozik az Intézet a lokális és táv-adatfeldolgozási — TAF — rendszerek fejlesztésével és telepítésével is.

A KERSZI „leányai"

Az elmúlt időszakban nagyfokú profilbővítést és egyidejűleg profiltisztítást is végrehajtottak a KERSZI-nél. Azokat a tevékenységeket, amelyek nem alapvetően meghatározóak az Intézet gazdálkodását és szakmai profilját, perspektíváját illetően, leányvállalati, illetve kft.-formában végzik. A három leányvállalat közül a KERCOMP kifejezetten irodatechnikára szakosodott. Tevékenységi körébe tartozik a vállalati átvilágítások elkészítése, új vállalkozó szervezetek menedzselése, rendszer- és ügyvitelszervezés, valamint az IBM-kompatibilis irodai rendszerek és ezek hálózatainak alkalmazása.

Merőben eltér a KERCOMP-étól egy másik leányvállalat, a NOVORG profilja, amely már nem kifejezetten és kizárólag számítástechnikai természetű vállalkozás. Vállalatszervezési kérdésekkel, illetve egyre inkább a decentralizálódó piaci igényekhez jobban igazodó kiadványok készítésével foglalkozik. Ez a tevékenység már kinőtte a kereskedelemnek mint ágazatnak a kereteit, és ágazatközi szintűvé vált. Annak idején például az adórendszer bevezetésénél a NOVORG jelentette meg azt az adótanácsadó naplót, amely lakossági forgalomba is került.

A harmadik leányvállalatnak — amely egy matematikai laborból nőtt ki — aztán már végképp alig van köze a számítástechnikához. A HUNGAROVIDEO — mert erről van szó — a kezdet kezdetén még abból indult ki, hogy nemcsak számítástechnikával és adatfeldolgozással áll a kereskedelmi vállalatok rendelkezésére, hanem reklámfilmek elkészítését is vállalja partnerei megrendelésére. Ez az üzlet olyan jól bejött, hogy 1988-ban már 80 millió forintot realizált a leányvállalat, s tevékenységi köre is egyre inkább áttevődött a lakossági szférába. Teljesen önállóan, önálló külkereskedelmi joggal végzi nyugati játékfilmek szinkronizálását és forgalmazását, természetesen nem a „fekete video" leple alatt, hanem annak a körnek a tagjaként, amely állami engedéllyel, licencekkel lényegében meghonosította Magyarországon a videózást.

A te(l)hetetlen (?) anya

Az egyetlen terület, amely még ma is hiányzik az anyavállalat tevékenységéből, a hardvergyártás, ezt az utóbbi két-három évben hardverkereskedelemmel igyekezett ellensúlyozni a KERSZI. Annak idején elindították a Commodore 610-es projektet — erről a Mikroszámítógép Magazin is hírt adott —, amely egy kétlépcsős program volt, illetve lett volna. Az első lépcsőben még nem is volt semmi gond: egy viszonylag rugalmas külkereskedelmi és ellenőrzési rendszer mellett sikerült gyorsan nyélbe ütni az üzletet. De sajnos a második lépcső a külkereskedelmi engedélyezés elhúzódása miatt csak nagyon későn léphetett életbe. Éppen belecsúszott abba az időbe, amikor egyéb csatornákból — főleg magánimportból — megjelentek Magyarországon az IBM (-kompatibilis) PC-k, és elárasztották a piacot. Uralkodó szerepük — mint köztudott — azóta is töretlen, így a 610-es projekt kudarcra ítéltetett.

Kereskedő, a piac minden rezdülését figyelemmel kísérő cég lévén természetesen a KERSZI sem vonta ki magát a PC-k forgalmazásából. Idén az Intézet is egy jelentősebb vállalkozásba kezdett, amelynek keretében viszonylag homogén — és ez nagy szó Magyarországon — PC-szállítmányt hoztak be az országba, és ezt forgalmazzák. A forgalmazás a KERSZI esetében nemcsak a gépeket jelenti, hanem a kapcsolódó szoftvert, szervizt és a garanciát is. Ugyanakkor — és ez nem mond ellent a piac eme szektora igényeinek — azt is világosan látják az Intézetnél, hogy a vállalati igények kielégítése PC-kkel és PC-hálózatokkal már egyre kevésbé oldható meg.

Arccal a mini felé

Ezért is fordult figyelmük egyre inkább a miniszámítógépes vonal felé. A Központi Fizikai Kutatóintézettel egy olyan együttműködési megállapodást kötöttek, amelyben a KFKI a kereskedelmi vállalati szférában egyfajta dealeri funkcióra jogosítja fel a KERSZI-t. Ez az értékesítésnél megfelelő kedvezményekkel is jár. Három ilyen rendszert terített a KERSZI az elmúlt egy évben, amelyek természetszerűleg nagyon komoly szoftverrendszer üzembe állítását is jelentik. A MERKUR-nál — ahol a vevő-, illetve személygépkocsi-nyilvántartást, valamint a garanciálisalkatrész-felhasználás rendszereit működtetik a mini-

Ez a KERSZI már nem az a KERSZI

számítógépeken — nem is annyira a feladat bonyolultsága, mint a tételszám nagysága indokolja ezt a kategóriát. Itt elsősorban a rendkívül rövid visszakeresési időknél jönnek ki a rendszer előnyei.

A Duna Füszért monori raktárházának számítógépes rendszere viszont — amely szintén KFKI-minigépen alapul — már egyértelműen nemcsak a tételszámban, hanem bonyolultságában és a funkciók összetettségében is komoly dolog. És ugyancsak nagyon komoly a tatabányai Centrum Áruháznál üzemelő, üzembe helyezésre kerülő, illetve fejlesztés alatt álló áruházi rendszer. Itt a VAX-alapú — TPA 520-as — rendszerhez Hugin Sweda pénztárgépek — mégpedig Master Slave üzemben működő pénztárgépek — csatlakozását kell, illetve kellett megoldani.

Ugyancsak ebben az évben állította üzembe a KERSZI az ÁPISZ Béke téri — Rodin — áruházának számítógépes rendszerét. Novell hálózaton zajlik a viszonteladók kiszolgálása, készletfigyeléssel, készletgazdálkodással. A rendszernek köszönhetően mind az áru-, mind pedig a pénzmozgás felgyorsult az áruházban.

A DOMUS áruház rekonstrukciójakor az információs rendszernek a rekonstrukciójára is sor került. Ennek keretében egy TPA 420-as — duálprocesszoros — rendszerrel oldották meg az áruház belső adminisztrációját, és a DOMUS Sipos Dénes utcai raktárának a táv-adatfeldolgozási vonalon történő összekötését. Ehhez a rendszerhez is csatlakozik IBM PC-n keresztül egy Hugin Sweda-alapú Master Slave rendszer. Itt is, akárcsak Tatabányán egy vonalkódos értékesítési rendszerről van szó, de míg Tatabányán még csak fejlesztés alatt van, a DOMUS-ban már üzemel is.

Vass Nándor, a KERSZI igazgatója elmondta, szeretnék, ha az Intézet neve mellől végképp eltűnne a „nagyszámítógépes bérmunkairoda" megjelölés; céljuk, hogy a KERSZI-t végre annak értékeljék, ami: teljes körű szolgáltatást nyújtó szoftverháznak. Ezt az is alátámasztja, hogy ahová a KERSZI szakemberei rendszerekkel mennek, ott üzemvitelt, üzemszervezést is vállalnak. Mindezt többféle variációban, formában.

Például a MOBIL Nagykereskedelmi Vállalatnál, a törökbálinti raktárban egy SZM 1420-as rendszerrel és IBM PC-kkel felépült hálózatot üzemeltetnek, két műszakban, mint az Intézet számítóközpontját, ugyanakkor a Belkereskedelmi Szállítási Vállalatnál a megrendelő számítástechnikai apparátusával együtt alakítottak korlátolt felelősségű társaságot, és így dolgoznak.

A megrendelők visszajelzése egyértelműen pozitív, ez főleg annak köszönhető, hogy nemcsak a gépet ajánlják, nemcsak a szoftvert szállítják, hanem a rendszer folyamatos üzemeltetését is vállalják. Ebben természetesen benne van a kockázat is, ha például egy winchester meghibásodik, a KERSZI-nek kötelessége azt haladéktalanul kicserélni.

Kötelező Forintot Termelni

A KESCH Kft.-t egy osztrák magáncéggel együtt alapította a KERSZI, ez alapvetően korszerű és professzionális perifériák kereskedelmével és rendszerbe illesztésével foglalkozik. E kft. révén az országban elsőként a KERSZI-ben üzemel az az ION sornyomtató, amely rendkívül jó minőségben és normál papírra készíti el a grafikus vagy alfanumerikus nyomtatványokat. A tetszőleges karakterkészlet révén mód nyílik akár kínai, japán vagy cirill karakterek megjelenítésére is. Nem véletlen, hogy a külkereskedelmi vállalatok, illetve az olyan cégek, ahol a nyomtatvány minősége, esztétikai megjelenése is fontos, egyre nagyobb számban keresik meg a KERSZI-t. A nyomtató alkalmas nagygépes adatok megjelenítésére és Novell hálózaton született anyagok nyomtatására is, illesztését az R 46-os géphez a KERSZI szakemberei oldották meg.

A másik vegyes vállalatnál — a Logistix Hungarynél — a Logistix cég svájci képviselete és a Hugin Sweda Austria a két külföldi partner. Ez a vegyes vállalat az OSZIR keretében beszállított System/36-os bázison telepített Logistix szállodai front-office rendszer installálására, betanítására és folyamatos felügyeletére alakult. Ez egy szoftverrendszer ugyan, de 15 helyen jelent a helyzet jelenlegi állásában installációt. Ami azt jelenti, hogy 15 HungarHotels-szállodában, illetve a Rege Szállóban kell biztosítania a vegyes vállalatnak a folyamatos működést.

A feladat jellege, nagysága olyan kvalitású — és bérigényű — szakemberek alkalmazását igényelte, amelyet a jelenlegi szabályozórendszer mellett a KERSZI hagyományos vállalati struktúrában nem tudott volna megoldani. Ezért is született meg ez a vegyes vállalat, s így lát garanciát a nyugati fél is arra, hogy a rendszer folyamatos üzemeltetésének feladatait el lehessen látni.

A Svájcban elkészült szoftverrendszernek a System/36-ra történő installálása egyidejűleg több feladatot is jelentett. Meg kellett tervezni az elhelyezést, a kábelezést, életre kellett kelteni magát a System/36-ost, majd adaptálni kellett a szállodai rendszert a magyar viszonyokra. Ezt követte még az adaptált rendszer betanítása és az esetleges — főként a rendszer nem-ismeretéből származó — hibák elhárítása.

Szerepel a tervek között a Logistix back-office rendszernek az elkészítése is.

Ez a KERSZI tehát már nem ugyanaz a KERSZI . . . Jelzi ugyanakkor, hogy a nagyvállalati létnek is megvannak a maga előnyei, az az eset, amely egy újabb vegyes vállalat alapításához kapcsolódik. Előrehaladott tárgyalásokat folytat a KERSZI a Philips Datasystem ausztriai céggel. A tervezett vegyes vállalat a Philips Datasystem meghatározott termékeire kereskedelmi képviseleti jogot kap, mindazokkal a garanciákkal, amelyek a dealeri funkciókra vonatkozó jogszabályok értelmében megilletik a vállalkozót. De a Philips Datasystem csak úgy volt hajlandó aláírni a megállapodást — tekintettel arra, hogy a KESCH Kft. az egyik partner —, ha a KERSZI mint anyavállalat ellenjegyzi azt. Tehát igaz, hogy a kisvállalkozások korát éljük, de azért nagyon fontos — főként nemzetközi piacban gondolkodva — egészséges egyensúlyt tartani a vállalkozásokban. A maga módján erre — is — törekszik a Kereskedelmi Szervezési Intézet.

BÖRZE

TUTTI

ELECTROCOOP KISSZÖVETKEZET

— Eredeti számítógépek
— Kompatibilis XT/AT, 386 konfigurációk
— Szünetmentes tápegységek
— Egyedi megrendelésfelvétel

Cím: Bp., Üllői út 81. 1091
Tel.: 334-354
Telex: 22-7230
Telefax: 149-869

„ENET"

Lokális hálózat
Ára gépenként
19 900 Ft+ÁFA

„SZÁK"

Számlakészítés —
ÁFA-nyilvántartás —
Könyvelés
Bérfeldolgozást is vállalunk.
EcoSoft Kisszövetkezet
Telefon: 181-2460

procontrol

IRÁNYÍTÁSTECHNIKA KISSZÖVETKEZET

MÁRKABOLT, BEMUTATÓTEREM
62/21-165, 28 985
Szeged, Kazinczy u. 8. Tel.: 12-259
Telex: 82-726

386 AT 25 MHz toronygép	299 000 Ft-tól
286 AT 16 MHz TURBO AT gép — 86 XT 12 MHz	98 000 Ft-tól
TURBO XT gép	69 000 Ft-tól
LAPTOP hordozható LCD XT gép + modem + akku	149 000 Ft
12" zöld monochrom monitor TUV M31	4 900 Ft
5 1/4" DS DD Floppy lemez	79 Ft

és még 2000 árucikkről árjegyzék, azonnali kiszolgálás.

BROTHER AX 15 típusú,
margarétafejes,
elektronikus írógép

Megvásárolható:
Budapest VI.,
Népköztársaság útja 2.
Tel.: 531-231

MIGÉRT

ASY ELEKTRONIKA

A legújabb ajánlatunkból:
— ASY—16 SZUPERMIKRO számítógépek (12 terminál, UNIX operációs rendszer)
— IBM PC/XT-AT-kompatibilis számítógépek
— megrendelő által definiált betűkészlettel rendelkező billentyűzetek
— elektronikai elemek (integrált áramkörök, ellenállások, tranzisztorok)
— monitordobozok, műszerházak
— szoftvertermékek és fejlesztések

KERESKEDELMI IRODA:
1061 BUDAPEST
LISZT FERENC TÉR 10.
TEL.: 415-166, TELEX: 22-4378

ECONORG SZÁMÍTÁSTECHNIKAI KÖZÖS VÁLLALAT

1146 Bp.,
AJTÓSI DÜRER SOR 10.
Levélcím:
1393 Pf.: 319.
Telefon: 421-974
Telex: 22-6544

SZAKÜZLETEK:

1. Bp. VI., Szinyei Merse u. 1.
Tel.:127-628. Tx.: 22-6684

2. 9022 Győr, Lukács S. u. 18.
Tel.: 96-14880. Tx.: 02-4679

3. 3100 Salgótarján, Ady Endre u. 1.
Tel.: 32-10971. Tx.: 22-9380

Új szolgáltatásokkal, új helyen várja kedves ügyfeleit az

ISKOLASZÁMÍTÓGÉP SZERVIZ

IBM és Commodore számítógépek javítása közületek és magánszemélyek részére

Éves átalánydíjas szerződések rendkívül kedvező feltételekkel.

Egyéb szolgáltatások:

— C16 bővítése 64 kbájtra,
— magyar ékezetes karakterkészlet beépítése
— játékprogramok eladása és vétele
— Tectronix oszcilloszkóp előnyös áron

A javítás ideje alatt szükség szerint cseregépet biztosítunk.

Címünk: 1088 Budapest, Rákóczi út 25. Tel.: 138-1121

Az Amiga programozása assembly nyelven I.

Az Amiga személyi számítógép különleges, nagy teljesítőképességű számítógép, szuper grafikával, hanggal és többfeladatos lehetőségekkel. Hardverét a Motorola cég MC68000-es mikroprocesszora köré építették fel, speciálisan erre a célra kifejlesztett VLSI áramkörök felhasználásával. Ezek az integrált áramkörök — a szakirodalom a „custom chips" névvel illeti őket — hozzák létre azt a grafikát és hangot, amely az Amigát oly sok számítógép-kedvelő számára teszi csábítóvá. Az Amiga rendszerszoftvere a 256 kbájt nagyságú ún. Kickstart ROM-ban található meg. Az 1.2-es verziójú Kickstart ROM-ban ebből 192 bájtnyi az effektív programterület. Az 1.3-as verzióban a programterület már jóval hosszabb lett.

Azoknak az Amiga-tulajdonosoknak, akik csak játék céljából vásárolták meg ezt a gépet, vagy akiknél a felhasználói programok jelentik az Amiga alkalmazásának csúcsát, fogalmuk sem lehet arról, milyen élvezet egy ilyen logikusan felépített rendszert assemblyben vagy C nyelven programozni. A C64 ROM-programjához képest ennek a gépnek a rendszerszoftvere merőben más szemlélettel érthető meg. Ez részben abból adódik, hogy az Amiga operációs rendszerét a feladat komplexitása miatt az assemblynél magasabb szintű nyelven — C-ben — írták meg. Ahhoz tehát, hogy mindazt kihasználhassuk, amire e számítógép képes, az MC68000-es mikroprocesszor assembly programozását vagy legalább a C nyelvet alapfokon ismernünk kell. E sorozat következő részeiben azonban nem programnyelveket szeretnénk tanítani — ezt megteszik helyettünk az ide vonatkozó és hazánkban is beszerezhető szakkönyvek. Céltudatosan olyan információkat kívánunk közkinccsé tenni, amelyek speciálisan az Amiga programozását segítik az alacsony szintű nyelveken: assemblyben és C-ben.

Miért csak az assembly nyelvet említjük cikkünk címében? Mert ez a programozók által használt legalacsonyabb szintű nyelv, éppen ezért ez igényli a környezet legmélyrehatóbb ismeretét. Ha tehát az assembly programozáshoz szükséges tudnivalókat (funkcióhívásokat, bizonyos fontosabb adatstruktúrákat stb.) adjuk közre, akkor ezek a C nyelvű programozó munkáját is ugyanúgy segítik. Itt szeretném hangsúlyozni, hogy a C és az assembly ezen a számítógépen szorosan összekapcsolódik. Egy assembly leírásból mindent megtudhat az, aki a C-t ismeri és fordítva: a C esetében használt include-fájlokból szinte minden lényegeset kihámoz az, aki ért az assemblyhez.

Az assembly lesz a középpontban, tehát elkerülhetetlen, hogy megismerkedjünk egy assemblerrel. Erre a célra a KUMA Computers K-SEKA nevű assemblerét tartom a legalkalmasabbnak. A K-SEKA egyáltalán nem mondható luxus követelményeket kielégítő fordítónak, mégis megbízhatósága, nagy sebessége (kb. 25 000 sor percenként), könnyű kezelhetősége miatt a legmegfelelőbb az első lépések megtételéhez, és a majdan ismertetésre kerülő példaprogramok működését is ezen garantálom. Az alábbiakban ennek az assemblernek a tömör ismertetését olvashatják.

A K-SEKA assembler parancskészlete

A szögletes zárójelek azt jelentik, hogy a paraméter megadása nem kötelező.

! Kilépés az assemblerből. Azonban mielőtt kilépne, a következő kérdést teszi fel nekünk: „Exit to System, Sure?" („Valóban ki akar lépni a rendszerbe?")

? kifejezés
Kiszámítja a kifejezés értékét.

@ paraméterek
A gépi kódú program futtatásakor az itt megadott paraméterek úgy funkcionálnak, mintha programunkat CLI-ben futtatnánk, és ezeket a paramétereket írtuk volna be a parancssorba.

A [kezdőcím]
Jelentése: „Assemble". Fordítás assembly nyelvről gépi kódra. Ha nem adunk meg kezdőcímet, akkor az előzőleg megszerkesztett programot fordítja le, egyébként pedig a kezdőcímtől elindulva, lépésenként adhatjuk be az assembly utasításokat. Ha a megszerkesztett forrásprogramot fordítjuk, és nem használtuk a LOAD pszeudo-utasítást, akkor a PC-be automatikusan betölti a tárgykód (a lefordított, gépi kódú program) kezdőcímét.

B
Jelentése: „Bottom of buffer". A szöveg végére állítja a kurzort.

C
Jelentése: „Copy memory". Tetszőleges memóriatartományt átmásol. A memóriatartomány paramétereit párbeszédes módon kell megadnunk. (BEGIN = a forrás-memóriablokk kezdőcíme. END = a forrás-memóriablokk vége. DEST = a cél-memóriablokk kezdőcíme).

D [szám]
Jelentése: „Down". A megadott számú sorral lefelé lépteti a kurzort. Az alapértelmezés 1 sor.

E [sorszám]
Jelentése: „Edit line". Szerkesztés céljából lehívja az aktuális vagy az itt megadott számú sort. Kilépés az ESC billentyűvel.

F
Jelentése: „Fill memory". Az általunk megadott bájttal feltölti a kijelölt memóriatartományt. A memóriatartomány adatait párbeszédes formában kell megadnunk. (BEGIN = a memóriatartomány kezdőcíme. END = a memóriatartomány vége. DATA = a feltöltésre használt bájt.)

G [cím]
Jelentése: „Go to address". A PC által mutatott címről vagy egy tetszőleges helyről elindítja a gépi kódú programot. A „BREAKPT >" promptjelnél töréspontokat állíthatunk be. Ha az általunk beállított töréspontok valamelyikéhez érkezik a program, akkor annak futása megszakad, és visszakapja a vezérlést az assembler. Az indítás JMP utasítással történik, ezért hibátlan visszatérést úgy hozhatunk létre a legkönnyebben, ha a visszatérés helyére egy ILLEGAL utasítást írunk.

H
Jelentése: „How big?" Ez a parancs az assembler pufferjeinek méretét íratja ki a képernyőre. Az assembler az alábbi pufferek kezdőcímét, végét és hosszát jeleníti meg:
Work munkaterület
Link a linker bemeneti pufferje
Src a forrásprogram (szöveg)
RelC a program relokációs táblája
RelD az adatok relokációs táblája
Code a kimeneti tárgykód kódszegmense
Data a tárgykód adatszegmense

I [sorszám]
Jelentése: „Insert line(s)". A kurzorsornál vagy egy tetszőleges helynél szúrhatunk be sorokat a forrásprogramba. Kilépés az ESC billentyűvel.

J [cím]
Jelentése: „Jump to address". Ugrás az általunk definiált vagy a PC által meghatározott címre töréspontok elhelyezése nélkül. Az ugrás JSR utasítással valósul meg, ezért programunkat egy RTS-nek kell lezárnia.

KS
Jelentése: „Kill Source". Törli a forrásprogramot, de előtte visszakérdez: „Sure?" („Valóban?")

KL
Jelentése: „Kill Link". Törli a linker bemeneti pufferjének tartalmát, de előtte még a biztonság kedvéért visszakérdez.

KF
Jelentése: „Kill File". Törli a megadott fájlt, de előtte ez is megerősítést kér.

L [szöveg]
Jelentése: „Locate text cursor". Az aktuális kurzorpozíciónál kezdve, a fájl vége felé haladva megkeresi a beadott szöveg első előfordulását, és ide állítja a kurzort. Ha nem adunk meg semmilyen szöveget, akkor az előző L parancsnál megadott szöveget keresi.

M [cím]
Jelentése: „Modify memory". A memóriatartalom bájtonkénti módosítására nyújt lehetőséget. A RETURN billentyű lenyomásával átléphetjük az aktuális bájtot, az ESC billentyűvel pedig kiléphetünk ebből a funkcióból.

N [cím]
Jelentése: „Disassemble". A beadott vagy az aktuális címtől kezdve assembly nyelvre fordítja vissza a gépi kódot.

O
Jelentése: „Old source text". Ha a KS (Kill Source) parancs után meggondoltuk magunkat, és vissza szeretnénk nyerni a törölt forrásprogramot, akkor ezt az O paranccsal megtehetjük. Ha a KS után akár csak egyszer átléptünk szerkesztési módba, többé már nem hozhatjuk vissza az elveszített programot!

P [szám]
Jelentése: „Print source text". Az aktuális sortól kezdve kilistázza az itt megadott hosszúságú programrészletet. RETURN helyett a CTRL+P billentyűkombinációt használva, a nyomtatóra kerül a lista.

Q [cím]
Jelentése: „Memory and ASCII dump". Memóriatartalom megjelenítése numerikus és karakteres formában, tetszőleges címtől kezdve.

R
Jelentése: „Read source text". Forrásprogramot olvas be a lemezről. A fájlt „.S" kiterjesztéssel keresi. (Lásd még: W parancs)

RI
Jelentése: „Read Immediate". Egy fájlban tárolt memóriatartományt visszatölt a kívánt helyre. (BEGIN = a memóriatartomány kezdőcíme. END = a memóriatartomány vége.) (Lásd még: WI parancs)

RL
Jelentése: „Read Link". Összeszerkesztésre alkalmas programmodult tölt be a linker pufferbe, de előtte még a biztonság kedvéért visszakérdez: „Sure?" („Valóban?") (Lásd még: WL parancs)

RO
Jelentése: „Read Object". Tárgykódot (futtatható gépi kódú programot) tölt be a memóriába. (Lásd még: WO parancs)

S
Jelentése: „Single step". A program lépésenkénti végrehajtása a PC által mutatott helyen.

T [sorszám]
Jelentése: „Target to line". A megadott számú sorra állítja a kurzort. Ha nem adunk meg sorszámot, akkor a kurzor az első sorba kerül.

U [szám]
Jelentése: „Up". Tetszőleges számú sorral felfelé lépteti a kurzort. Az alapértelmezés 1 sor.

V [útvonal]
Jelentése: „View disk". Kilistázza az útvonalon található fájlokat.

W
Jelentése: „Write source text". Elmenti a pufferben lévő forrásprogramot. Az általunk megadott fájlnév „.S" kiterjesztést kap a lemezen. (Lásd még: R parancs)

WI
jelentése: „Write Immediate". Tetszőleges memóriatartomány mentése. (BEGIN = a memóriatartomány kezdőcíme. END = a memóriatartomány vége.) (Lásd még: RI parancs)

WL
Jelentése: „Write Link". Összeszerkesztésre alkalmas programmodult ment el. A fájl nevéhez automatikusan hozzáadja a „.LNK" kiterjesztést. Ez a parancs csak akkor használható, ha a fordításnál (A parancs) L opciót adtunk meg! (Lásd még: RL parancs)

WO
Jelentése: „Write Object". A lefordított, tehát gépi kódú programot menti el. A fájlnév nem kap semmilyen kiterjesztést. (Lásd még: RO paracs)

X [regiszter]
Jelentése: „eXamine registers". Az X parancs önmagában az MC68000-es regisztereinek megjelenítésére szolgál. Ha az X után valamelyik regiszter nevét írjuk be, akkor annak tartalmát módosíthatjuk.

Z [szám]
Jelentése: „Zap (delete lines)". Az aktuális kurzorpozíciótól kezdve az általunk megadott számú sort törli. Az alapértelmezés 1 sor.

Direkt módon megvalósítható szövegszerkesztő funkciók

ESC
Parancs üzemmódot vagy szerkesztési üzemmódot állít be.

CTRL + B
Megjelöli a másolandó blokk kezdetét.

CTRL + C
A blokk kezdete és az aktuális kurzorpozíció közé eső szövegrészt kivágja a szövegből, és tárolja az átmeneti tárban.

CTRL + H
A kurzortól balra egy karaktert töröl.

CTRL + I
Egy tabulátorpozícióval jobbra viszi a kurzort.

CTRL + M
Hatása megegyezik a RETURN billentyűével.

CTRL + O
Új sort szúr be a szövegbe.

CTRL + P
A CTRL + C billentyűkombinációval kivágott szövegrészt beszúrja a kurzor helyénél. FIGYELEM! A CTRL + P-nek van egy másik felhasználása is. Ha a ki-

adandó parancs kiviteli eszközéül a nyomtatót szeretnénk használni, akkor a sort nem a RETURN billentyű lenyomásával, hanem a CTRL + P billentyűkombinációval kell beadnunk.

Az „A" (assemble) parancsnál megadható fordítási opciók

H

Egy lap kiírása után megáll, és egy tetszőleges billentyű megnyomásáig várakozik.

L

Összeszerkesztésre (linking) alkalmas kódot generál. Ez után az opció után használhatóvá válik a WL (Write Link) parancs.

O

Optimalizálja a „branch" utasításokat.

P

A nyomtatóra viszi ki a kód és a szimbólumok listáját.

V

A képernyőre listázza a kódot és a szimbólumtáblát.

A forrásprogramban használható pszeudo-operátorok

Emlékeztetőül: a pszeudo-operátorok olyan, a forrásprogramban elhelyezendő utasítások, amelyek nem tartoznak a processzor utasításkészletébe, csupán a fordító működését befolyásolják.

; megjegyzések

A pontosvesszőtől a sor végéig terjedő részt figyelmen kívül hagyja a fordító.

ALIGN x

A tárgykód ez után következő részét x=2 esetén szóhatárra, x=4 esetén dupla szóhatárra igazítja. (Használhatunk más számokat is, de ez utóbbiakra van szükség a leggyakrabban.)

BLK.L x,y

BLK.W x,y

BLK.B x,y

„x" elemű memóriablokkot foglal le, amelyet az „y" kezdőértékkel tölt fel. Az elemek mérete „.L" esetén négy bájt (Long word), „.W" esetén két bájt (Word), „.B" esetén egy bájt (Byte).

CODE

Relatív fordítási üzemmódba kapcsol át.

DATA

Az adatszegmens kezdetét jelzi. (Az utána álló programrész nem hoz létre tárgykódot.)

DC.L x1,x2,x3, . . .

DC.W x1,x2,x3, . . .

DC.B x1,x2,x3, . . .

„.L" esetén duplaszavakat, „.W" esetén szavakat, „.B" esetén bájtokat foglal le a memóriában. „x1" és társai a lefoglalt egységek kezdőértékei, a felsorolt kezdőértékek száma tudatja az assemblerrel, hogy hány elem számára kell helyet foglalni.

END

A forrásprogram végét jelző kulcsszó.

EVEN

A tárgykód ez után következő részét páros memóriacímen kezdi az assembler. (Lásd még: ODD)

EQU vagy =

Szimbólumdefiniálást végző, egymással egyenértékű pszeudo-utasítások. A Maximum: EQU $20 teljesen azonos hatású a Maximum = $20 utasítással, mivel mindkettő a hexadecimális 20-as értéket rendeli hozzá a „Maximum" szimbólumhoz.

GLOBL szimbólum1, szimbólum2

Ha a teljes programot — például annak nagy terjedelme miatt — modulokból állítjuk össze (linking), akkor ezzel az utasítással a felsorolt szimbólumokat láthatóvá tehetjük minden modul számára. Ezt a pszeudo-utasítást mindig annak a fájlnak az elején kell elhelyezni, amelyben a szimbólumokat definiáljuk.

IF kifejezés

.

assembly program

.

ELSE

.

assembly program

.

ENDIF

Az IF kulcsszó után álló logikai kifejezést az assembler *a fordításkor* kiértékeli. Ha a kifejezés igaz, akkor az IF után következő, ha nem, akkor az ELSE után következő programrészletet fordítja be a tárgykódba.

ILLEGAL

Egy 68000-es utasításkészletébe nem tartozó „utasításkódot" fordít be a programba. Végrehajtása a $4 számú kizárást (exception, Guru Meditation stb.) hozza létre.

LINE_A x

A 68000-esen nem implementált utasítást hoz létre, amely a $A számú kizárást (exception) eredményezi. Például a „Line_A $FFF" utasításnak a $AFFF felel meg.

LINE_F x

A 68000-esen nem implementált utasítást jelenti, amely a $B számú kizárást (exception) hozza létre. Például a „LINE_F $123" utasításhoz a $F123 kód tartozik.

LIST

Engedélyezi a listázást fordítás közben. A LIST után az alábbi betűk állhatnak:

c: makróhívások listázása

d: makródefiníciók listázása

e: a makrókról is fordítási listát készít.

(Lásd még: NLIST)

LOAD kezdőcím

A tárgykód kezdőcímét adja meg a fordításhoz. Csak akkor használható, ha ORG-gal megadjuk a címkék kezdőcímét is.

név: MACRO

assembly program

.

ENDM

Makró definiálása. A „név" a makró neve. A MACRO sorától az ENDM soráig leírt sorok alkotják a makró törzsét. A makró meghívása során (ez nevének leírásával történik) a fordító bemásolja a kódba a makró törzsét. Legyen például egy makrónk, amely két számot összead:

```
Sum: MACRO
     move.1. ?1,d0
     addi.1. ?2,d0
     ENDM
```

Ezt a makrót a következőképpen hívjuk meg:

Sum: #$20, #$50

A makró a „?1" szimbólumba átveszi a #$20 számot, a „?2"-be pedig a #$50-et. A kérdőjel után álló szám azt mondja meg, hogy a makró hívásakor átadott paraméterek közül hányadikról van szó. Különleges szerepe van a makrókban használható „?0" szimbólumnak (változónak). Ez „001"-ről indul, és minden makróhívás során eggyel nő az értéke.

NLIST

Tiltja a listázást fordítás alatt. (Lásd még: LIST)

ODD

A tárgykódot páratlan memóracímen folytatja az assembler. (Lásd még: EVEN)

ORG kezdőcím

A címkék kezdőcímét adja meg a fordításhoz. (Lásd még: LOAD)

PAGE

A fordítási listázásnál lapozást eredményez.

PINIT adat1, adat2,...

A nyomtatót inicializáló adatsorozat megadása.

PLEN=hossz

A kinyomtatott lapok hosszának megadása.

PWID=szélesség

A nyomtatott lapok szélességének definiálása.

A forrásprogramban használható a

1. lista

```
ExecBase:           equ     4
OpenLibrary:        equ     -$228
CloseLibrary:       equ     -$19e
SetRGB4:            equ     -$120

start:      move.l    ExecBase,a6       ;az EXEC báziscíme
            move.l #0,d0                ;0-ás verziószám
            lea GfxName,a1              ;pointer a könyvtárnévre
            jsr OpenLibrary(a6)         ;a GRAPHICS megnyitása
            beq NoGfx                   ;ha nem tudta megnyitni..
            move.l d0,GfxBase           ;a báziscím eltárolása

            move.l #0,d0                ;0-ás verziószám
            lea IntuitionName,a1        ;a könyvtár neve
            jsr OpenLibrary(a6)         ;az INTUITION megnyitása
            beq NoIntuition             ;ha nem sikerült...
            move.l d0,IntuitionBase     ;a báziscím elmentése

            move.l IntuitionBase,a0     ;az INTUITION báziscíme
            move.l 56(a0),Screen        ;az IntuitionBase+56 cí-
                                        ;men találjuk meg az
                                        ;aktív screen címét

            move.l Screen,a0            ;az aktív screen
            addi.l #44,a0               ;ViewPort struktúrájának
            move.l a0,ViewPort          ;kezdőcíme

            move.l GfxBase,a6           ;a GRAPHICS báziscíme
            move.l ViewPort,a0          ;az aktív ViewPort
            move.l #0,d0                ;0-ás palettaszín
            move.l #$f,d1               ;R(vörös)=15
            move.l #0,d2                ;G(zöld)=0
            move.l #0,d3                ;B(kék)=0
            jsr SetRGB4(a6)             ;beállítja a palettaszínt

            move.l ExecBase,a6          ;az EXEC báziscíme
            move.l IntuitionBase,a1     ;az INTUITION báziscíme
            jsr CloseLibrary(a6)        ;bezárja az INTUITION-t

NoIntuition: move.l GfxBase,a1          ;a GRAPHICS báziscíme
            jsr CloseLibrary(a6)        ;bezárja a GRAPHICS-ot

NoGfx:      illegal                     ;biztonságos kilépés
                                        ;kizárással (exception)

GfxName:            dc.b    "graphics.library",0
even
IntuitionName:      dc.b    "intuition.library",0
GfxBase:            dc.l    0
IntuitionBase:      dc.l    0
Screen:             dc.l    0
ViewPort:           dc.l    0
```

négy alapművelet (+, —, *, /), valamint a logikai VAGY (jele: !), a logikai ÉS (jele: &) és a logikai KIZÁRÓ-VAGY (jele: ~). A kiértékelés mindig balról jobbra történik! Ahol számoknak kell szerepelniük, ott használhatók a forrásprogramban definiált szimbólumok, címkék. A K-SEKA alapértelmezés szerint decimális számrendszert feltételez! A hexadecimális számrendszer jele: $, az oktálisé: @, a binárisé: %.

Az operációs rendszer könyvtárai

Az Amiga operációs rendszere jól elkülöníthető rutinokból (funkcióból) áll, amelyeket feladataik alapján különböző könyvtárakba csoportosítottak. Például a lemezkezeléssel kapcsolatos műveleteket a DOS könyvtár, a memória és a feladatok kezelését az EXEC könyvtár, a grafikát a GRAPHICS könyvtár, a programozó egyéni igényét kielégítő képernyők és ablakok létrehozását az INTUITION nevű könyvtár rutinjai végzik el. Alapszabály, hogy a rutinok meghívásakor a D0, D1, A0 és A1 regiszterek tartalma nem marad sértetlen, vagyis az esetek többségében átíródik a tartalmuk. Ez főleg abból ered, hogy rajtuk keresztül adunk át paramétereket, és bennük jelennek meg a funkció eredményei. Az összes többi regiszter tartalma nem változik meg.

A processzor A6 regisztere igen fontos szereppel bír ebben a rendszerben. Amikor egy könyvtári rutint meghívunk, ennek a címregiszternek az illető könyvtár kezdőcímét kell tartalmaznia. Egy funkciót úgy hívunk meg, hogy megadjuk a funkció és a könyvtár kezdőcímének ofszetjét (különbségét). Éppen ez a relatív címzésmód (egészen pontosan: „címregiszter indirekt, ofszettel" címzésmód) teremti meg a különböző ROM-verziók közötti kompatibilitást.

Legelső feladatunk, hogy megnyissuk a használni kívánt könyvtárat. A megnyitást végző rutin a D0 regiszterben adja vissza a könyvtár kezdőcímét. (Ha a D0-ban 0-át kapunk vissza, az azt jelenti, hogy nem sikerült az akció.) Viszont az „OpenLibrary" funkció szintén egy könyvtár részét képezi, mégpedig az EXEC-ét. Hogyan tudjuk meg akkor az EXEC báziscímét? A megoldás egyszerű. Az EXEC kezdőcímét a $4 címen találjuk meg — ez tulajdonképpen a 68000-es PC-jének kezdőértéke RESET esetén. Az „OpenLibrary" funkciót a következő assembly utasítással hívjuk meg:

jsr OpenLibrary (a6)

Ahhoz, hogy ezt értse az assembler, definiálnunk kell az „OpenLibrary" szimbólumot mint a rutin ofszetjét:

OpenLibrary: equ -$228

A rutin valódi belépési címe az A6 regiszter tartalmának és a -$228 értéknek az előjelhelyes összegzésével adódik. Az „OpenLibrary" bemeneteként A6-ba a $4-es címen található címet kell tölteni, D0-nak a verziószámot kell tartalmaznia, A1-nek pedig a könyvtár nevére kell mutatni, amely egy a RAM-ban elhelyezett sztring (például: „graphics. library"). Programunk befejeztével a nyitott könyvtárakat is be kell zárni. Erre a „CloseLibrary" rutin szolgál. Ez is az EXEC-ben található, tehát A6-ba az EXEC báziscímét (a $4-es címen található) kell tölteni. A1-nek most a bezárni kívánt könyvtár báziscímét kell tartalmaznia. A hívás a következőképpen történhet:

jsr CloseLibrary (a6);

ahol a „CloseLibrary" szimbólumnak -$19e az értéke.

C nyelven mindez sokkal tömörebben elvégezhető. Például a GRAPHICS könyvtár megnyitása így néz ki:

GfxBase=OpenLibrary („graphics. library", 0);

ahol a „GfxBase" változónak long típusúnak kell lennie. A könyvtárzárás is sokkal egyszerűbb:

CloseLibrary (GfxBase);

Befejezésképpen egy rövid programmal illusztrálom a könyvtárhasználat módját *(1. lista)*. A program a GRAPHICS könyvtár SetRGB4 nevű funkciójának segítségével megváltoztatja a képernyő 0-ás palettaszínét, vagyis a háttérszínt. Az összehasonlíthatóság kedvéért ezt megírtam C-ben is *(2. lista)*.

2. lista

```
#define RED    0xf
#define GREEN  0
#define BLUE   0
#define COLOR  0

#include "exec/types.h"
#include "intuition/intuition.h"
#include "intuition/intuitionbase.h"

long GfxBase=0;
long *vp;

main()
{
     struct IntuitionBase *IntuitionBase;
     struct Screen *screen;

     GfxBase=OpenLibrary("graphics.library",0);
     if (GfxBase==NULL)
     {
          exit(1000);
     }

     IntuitionBase=OpenLibrary("intuition.library",0);
     if (IntuitionBase==NULL)
     {
          CloseLibrary(GfxBase);
          exit(2000);
     }

     screen=IntuitionBase->ActiveScreen;
     vp=&screen->ViewPort;
     SetRGB4(vp,COLOR,RED,GREEN,BLUE);

     CloseLibrary(IntuitionBase);
     CloseLibrary(GfxBase);
}
```

Tóth Zoltán

A Solarsoft kínálatából

Mint arról már 1989/11-es számunkban hírt adtunk, új szoftverkategória jelentkezett a magyar piacon is: a shareware. A Cédrus Kisszövetkezet Magyarországon elsőként vállalkozott arra, hogy a vékony pénztárcájú, PC-hez, PC-kompatibilis géphez csak alkalomszerűen hozzájutó, tanulni, illetve ötletet meríteni kívánó felhasználók számára is megfelelő árú szoftvertermékeket forgalmazzon. Úgy véljük, olvasóink között is sokan akadnak, akik csak ezt a szoftvert tudják megfizetni, ezért határoztuk el, hogy rendszeresen közzétesszük a választást, eligazodást megkönnyítő Solarsoft-katalógus információit.

Név:
COLORADO ENTERPRISES TURBO C TUTOR V. 1.5
Szerző:
Gordon Dodrill, New Mexico, 1987.
Leírás:
A Turbo C nyelv tanulásához ad segítséget a közel 80 példaprogramból (.C) és rövid magyarázatokból (CHAP*.TXT) álló gyűjtemény.*
A fontosabb témakörök:
— programok felépítése
— vezérlőszerkezetek
— függvények, függvényhívások
— definíciók, makrók
— sztringek, tömbök
— mutatók (pointerek)
— I/O műveletek
— fájl I/O műveletek
— struktúrák
— memóriaterület-foglalás
— karakteres és bitműveletek
— egyéb példák
— Visual Calculator
C-ben írt program, amellyel szinte programsoronként beírt, illetve tárolt adatokon számítások végezhetők. Saját kis HELP-je van, a VC.DOC pedig egy rövid oktató leírás.
Alkalmazható függvényei:
ABS(), SQRT(), EXP(), LOG (), SIN(), COS(), ATAN(), FACT().
Dokumentáció:
A COMPILER.DOC fájl a fordítók installálását, az 1 vagy 2 floppy, illetve a harddiszk használatát, az első program írását és indítását mutatja be a Turbo C 1.0 és 1.5 verziók alapján.
Melléklet: printelőprogram, amellyel a C-programokat és a százoldalas magyarázószöveget lehet kiíratni.
Konfiguráció:
—

Név:
COLORADO ENTERPRISES MODULA-2 TUTORIAL V. 1.00
Szerző:
Gordon Dodrill, New Mexico, 1987.
Leírás:
A Modula-2 nyelv tanulásához ad segítséget a közel 72 példaprogramból (.MOD) álló gyűjtemény, rövid kiegészítő magyarázatokkal (CHAP*.TXT). A Pascalhoz hasonlítható.*
A fontosabb témakörök:
— alapok
— egyszerű adattípusok
— vezérlőszerkezetek
— eljárások (procedure)
— tömbök, típusok, állandók, címkék
— eljárások egymásba ágyazása
— I/O műveletek (monitor, billentyűzet, fájl, printer)
— példaprogramok: SCALARS, SUBRANGES, SETS
— rekordok
— mutatók, memóriaterület-foglalás
— modulok (local, global)
— géptől függő lehetőségek
— konkurencia
— egyéb példák
Dokumentáció:
A COMPILER.DOC fájl a Modula-2-verziók fordítóhasználatát mutatja be, néhány eltérés ismertetésével:
— Logitech Modula-2/86 2.00
— FTL Modula-2 1.20
— PCollier 1.00
Melléklet: printelőprogram, amellyel a Modula-2 százoldalas magyarázószövegét és a forrásprogramokat lehet kiíratni. (READ.ME, PRINTALL.BAT, LIST.EXE).
Konfiguráció:
—

Név:
TURBO PASCAL MULTI-TASKING SUBSYSTEM V. 1.10
Szerző:
Christian Philipps, 1988.
Leírás:
Az 50 DEM-be kerülő Multi-Tasking Subsystem lehetővé teszi Turbo-Pascalban a multiprogramozást.
— 50 párhuzamos tasklehetőség
— időszeletek (a nagyság programozható)
— üzenetközvetítés
— szemaforok
— 2.11 DOS-tól fölfelé használható
— 5 1/4", illetve 3 1/2" floppykezelés
A multiprogramozáshoz szükséges részletes Pascal nyelvű lehetőségek felsorolását is tartalmazza.
Dokumentáció:
A TP4MULTI.DOC német nyelvű ismertetés.
Konfiguráció:
—
Fontos:
A lemezen csak a Pop — Unit v. 1.10 DEMO található

Név:
TURBO BASIC TOOLS
Szerző:
Public Domain Software, Guido Ahlswede, Lerchenweg, 1989.
Leírás:
8 db Turbo-Basic forrás, modul és .EXE program

— dátumellenőrzés	*(DATUM1)*
— dátumellenőrzés, kiírás	*(DATUM2)*
— szövegellenőrzés	*(EINGABE)*
— futó szöveg	*(LAUF)*
— menü	*(MENUE1)*
— menü + ellenőrzés	*(MENUE2)*
— display feltöltése karakterekkel	*(TCLS)*
— ablak	*(WINDOW)*

Dokumentáció:
Német nyelvű kommentárok a forrásfájlokban.
Konfiguráció:
—

Név:
ADVBAS.LIB V. 4.0
Advanced Function Library for BASIC Compiler
Szerző:
Hammerly Computer Services, Inc. (HCSI)
Thomas Harlin, USA (Washigton, Alexandria), 1987.
Leírás:
230-féle assemblerkiegészítés a BASIC használatához.
A LIBRARY.ARC kibontása után 3 fő fájlt kapunk:

— ADVBAS.EXE	*QBASIC 2.0—3.0*	*— RUN-hoz*
— ADVBAS.QLB	*QBASIC 4.0*	*— RUN-hoz*
— ADVBAS.LIB		*— .EXE fájl előállításához.*

A SOURCE. ARC az assembler forráskódokat (.ASM) és 3 demóprogramot (*.BAS) tartalmaz.*
*Az OBJECT.ARC az *.OBJ-eket tartalmazza.*
Használata QB 4.0 esetén:
Az ADVBAS .QLB és .LIB állományainak a QBASIC könyvtárba való bemásolása után — QB /L ABVBAS-sal, illetve
— QB [programnév] /L ADVBAS-sal indul.
QB 2.0—3.0 esetén:
Az ADVBAS .EXE és .LIB állományok bemásolása után
— QB /L ADVBAS-sal, illetve
— QB [programnév] /L ADVBAS-sal indul.
Dokumentáció:
A 230 kiegészítés részletesebb leírását az ADVBAS.DOC, a rövid ismertetőket pedig az ADVBAS.QRF állomány tartalmazza.
Konfiguráció:
—

Mobilitási függvények

A sakkprogramok értékelőfüggvényében jelentős szerepet tölt be a mozgékonysági érték, amit három különböző definíció alapján értelmezhetünk:

1. A legális lépések száma.
2. A pszeudolegális lépések száma. Ezek olyan lépések, amelyeket megtehetnénk a táblán, ha nem vennénk figyelembe a sakk lehetőségét, vagyis azt, ha királyunk sakkban marad, miután egy kötésben lévő figurával elléptünk a helyéről.
3. De Groot-lépések. Ezt a definíciót De Groot professzor adta, aki a függvényét a következőképpen határozta meg: mobilitáson a pszeudolegális lépések számát értjük, ha az egyik fél királya sincs sakkban. Ellenkező esetben — vagyis ha bármely király sakkban van — a függvényt nem értelmezzük.

Nagyon érdekes eredményre jutunk, ha összehasonlítjuk Eliot Slater elméleti függvényét Dap Hartman gyakorlati eredményével. Az *1. ábrán* a háromszögeket összekötő vastag vonal az elméleti függvényt mutatja, a vékony folyamatos és a vékony szaggatott vonal pedig a kiválasztott 832 nagymesterjátszma analizálásával kapott görbe. Az ábrán jól látható, hogy az elméleti és a gyakorlati érték a 65. féllépésig jól megközelíti egymást.

Az analizált 832 nagymesterjátszmából érdemes külön megvizsgálni azokat, amelyek a 40. és külön azokat, amelyek a 30. lépés előtt fejeződtek be. Ezeket szemlélteti a *2.* és a *3. ábra.*

A következő táblázatban a játszmát tíz különböző szakaszra, intervallumra osztottuk tíz féllépésenként. A K_i oszlop az átlagos mozgékonysági értéket mutatja, a K_{max} a mozgékonysági érték maximumát, a K_{min} pedig a mozgékonysági érték minimumát.

Intervallum (féllépés)	K_i	K_{max}	K_{min}
3—8	29,1	38	22
9—14	34,3	47	25
15—20	35,8	50	26
21—30	37,1	52	24
31—40	39,4	57	25
41—50	38,0	65	21
51—60	34,0	50	23
61—80	29,1	43	12
81—100	23,1	42	10
100— . . .	20,3	36	7

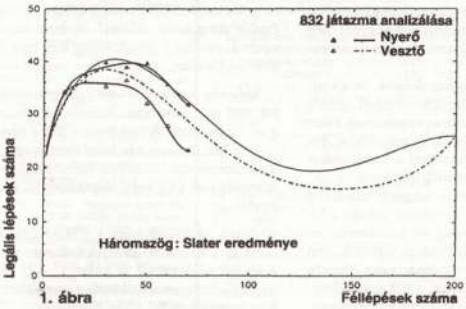
Érdemes összehasonlítani a három különböző értelmezés szerinti mobilitási érték átlagait a játszma különböző szakaszaiban.

Intervallum (féllépések száma)	Legális lépések számának átlaga	A pszeudo-lépések számának átlaga	De Groot-féle lépések számának átlaga
1—20	32,27	32,60	32,43
21—30	38,55	38,99	38,83
31—40	39,90	40,53	40,26
41—50	39,60	40,59	40,16
51—60	37,70	38,84	38,28
61—80	32,75	34,73	33,64
81—100	25,51	28,46	26,60
101— . . .	20,66	24,13	21,80

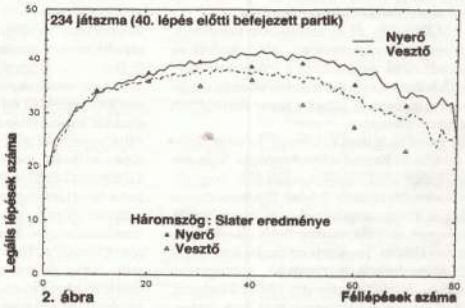
A *4., 5.* és a *6. ábrán* látható függvények olyan érdekes összefüggéseket tárnak fel, amelyeket a sakkprogramkészítők és a sakkjátékosok egyaránt hatékonyan felhasználhatnak.

Ezzel a sakkprogramkészítésről szóló sorozatunkat befejeztük, de igérjük, hogy ha jelentősebb előrelépés történik a sakkalgoritmusok elméletében, visszatérünk a témára.

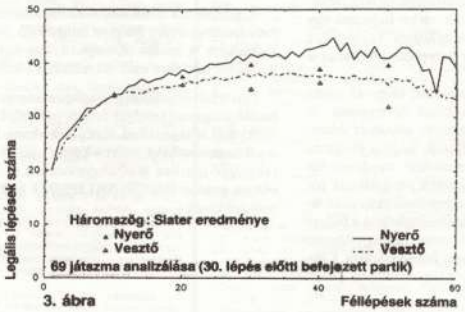
Kovács P. Attila



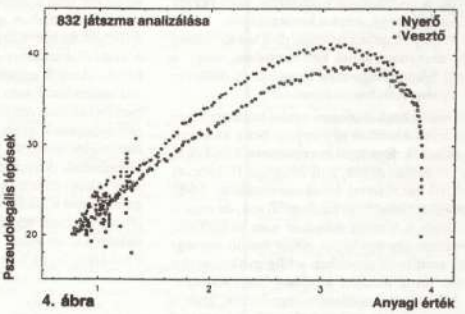
1. ábra



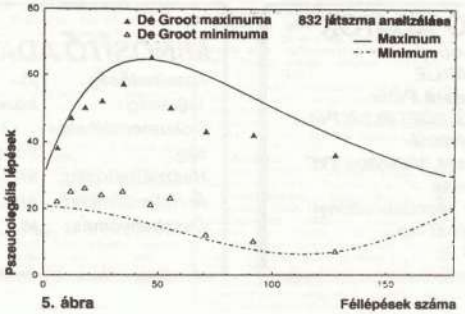
2. ábra



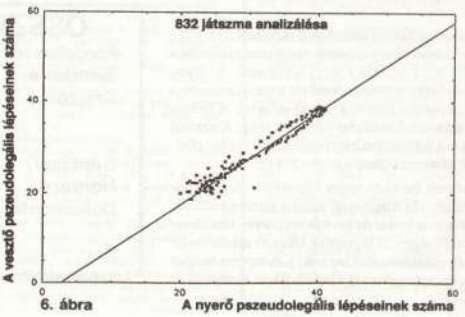
3. ábra



4. ábra



5. ábra



6. ábra

Ismét egy tesztgeneráló program

Ahhoz, hogy a számítógépek ténylegesen be tudjanak vonulni az oktatásba, az kell, hogy a pedagógusok szerszámként tudják őket használni. Ehhez olyan programokra van szükség, amelyek nem tételeznek fel túl mély számítógépi ismereteket. Két ilyen programot már elemeztem egyszer, és megállapítottam róluk, hogy ebben a szellemben beleillenek a pedagógusok szerszámosládájába. Most egy hasonló célokkal írt programot vizsgáltam meg. Nézzük, hogyan teljesíti az elvárásokat. A program fő jellemzőit a szokásos táblázat tartalmazza.

A CHARLIE teszt- és drillfeladatok készítését, belövését, tárolását, visszatöltését, kikérdezését és gyakoroltatását oldja meg. Ehhez a leírásában ígérteknek megfelelően tényleg minimális számítástechnikai ismeretet igényel. A gépet a tanár lényegében írógépnek használhatja.

Egy jó program még nem ér semmit, ha nem párosul hozzá megfelelő használati technológia. Sajnos a CHARLIE éppen attól szenved leginkább, hogy alkalmazási technológiájának leírása finoman fogalmazva hiányos. Ezt nemcsak az okozza, hogy a leírás csupán egylapnyi, és pillanatokon belül elkallódhat, hanem az egyébként programozástechnikailag jól megoldott online help is magyartalan, szövegezése pedig homályos. Emiatt a programot fel kell fedezni, ami egy kevésbé merész tanárnak már nem biztos, hogy sikerül. A technológia felfedezése után persze már minden úgy megy, mint a karikacsapás.

Az online helpen nehéz javítani, de a leíráson még lehet. Egy avatatlannak el kell mondani, hogy a programmal feleletkiválasztós teszteket és drilleket lehet írni. A témakör bármi lehet.

A nyelvi teszteket két idegen nyelvi karakterkészlet is segíti. Ezek azonban egy teszten belül keverten nem használhatók. Egy teszt maximálisan 8 lapból és laponként 24 sorból állhat, ami összesen 192 sor. A soronkénti 40 karakterrel tehát maximálisan 7680 karakteres tesztet lehet írni. Ez nem túl sok, de nem is kevés. A tanár a tesztek írásakor arra számítson, hogy a program egy-egy lapot olyan önálló egységnek tekint, amit tesztmenetben addig gyakoroltat a delikvenssel, amíg minden kérdésre helyesen nem válaszol. A lapok bekezdésekre tagolhatók, amit a bekezdés végén beírandó új bekezdés jel határol (SHIFT + =).

A szövegbe a helyes válaszokat is be kell írni. Ezeket megkülönböztetés céljából szögletes zárójelben ([=SHIFT + :,] =SHIFT + ;) kell tenni. Az ilyen szövegek helyére a tesztmenetben a program annyi pontot ír, amennyi helyet a válasz elfoglal. A helyes válaszok más színű betűkkel jelennek meg. A nebuló csak azokra a válaszokra kap pontot, amelyeket elsőre eltalál. Minden válasz egy pontot ér.

A feladatok bevitele teljes képernyős üzemmódban történhet. Használhatók tehát a kurzorgombok. Ez főként a már beírt, de javításra szoruló részek eléréséhez szükséges. A lapoknak lehet és ajánlatos címet adni (legalább az első lapnak!). A lapcím megjelenik a menüben, valamint a betöltéskor. Betöltéskor ennek alapján dönthetünk, hogy ezt a tesztet akarjuk-e betölteni (RETURN), vagy tovább keresünk a kazettán (bármilyen más billentyű). A beírható kérdések száma nincs korlátozva. Csak a lapok sormérete és a lapok száma korlátoz. Így a kérdések maximális hossza 24 sor. Feleletkiválasztós teszteknél kérdésenként akárhány soros akárhány választ meg lehet adni. Egy-egy kérdés—válasz tétel azonban nem nyúlhat át egy következő lapra, mert azt a program zavaró módon szétvágná a tesztmenetben. A 8 lapba maximálisan 20—40 feleletkiválasztós kérdést lehet belepréselni. A technológiai tudnivalókat ezzel még mindig nem merítettem ki teljesen, de nem is nekem kellett volna megírnom.

A bevitelnél vannak szokatlan dolgok. A Commodore gépeknél sajnálatos módon meglévő, idézőjelekkel kapcsolatos cirkuszokat a programozó kikerülte ugyan, de más nyűgöket adott ajándékba. Ha olyan helyekre megyünk a kurzorral a beírás teljes képernyős kezelése során, ahol még nem jártunk, akkor a bevitt szöveg a visszaíráskor meglepő képet fog mutatni. A program ugyanis tömörítve tárolja a kiírandó szöveget, kiíráskor pedig 40 karakteres sorokra tördel. A tördelés szóhatárokon történik, ami szép dolog, de hosszabb szavak összeesése jelentős veszteségeket okozhat a kiíráskor, mert az elválasztójelre a tördelő nem reagál. A korábban beírt lap javításakor igen kellemetlen a tömörített szöveg látványa. Ha korábban igyekeztünk térbe helyezni egy szöveget, az a javításkor már nem látszik. Használjuk a SPACE karaktert az olyan helyeken, ahol csak a helyet akarjuk lefoglalni.

Feleltetésnél nem lehet megadni, hogy az egyes osztályzatokra milyen ponthatárok érvényesek. A pedagógusnak ezért megmarad az unalmas adminisztrációs munka. További bánat, hogy a program alapelvéből következően semmiféle randomizálás nem vihető be a tesztekbe, hiszen a programnak fogalma sincs a kikérdezendő anyag logikai struktúrájáról. A nebuló így aztán hamar beszajkózza a helyes válaszokat annyira, hogy túljusson egy-egy lapon. Puskázhat is, ha a helyes választ felírja papírra. Olykor azonban erre sincs szükség, mert ismétlődő menet előtt a program nem mindig törli ki a képernyőablakot, ezért a helyes válasz leolvasható a képernyő tetejéről.

További meglepetés a kezdőknek a RESET gomb. Az egylapos leírás már a szöveg elején lelkesen magyarázza, hogy a RESET gomb megnyomásával a főmenű mindig azonnal visszatér. Vannak, akik mindent rögtön kipróbálnak, ami a leírásban szerepel. Sajnos én is így szabadultam meg először a program betöltésekor még meglévő, de a leírásban már meg sem említett demo-anyagtól. A RESET gomb ugyanis törli mind a 8 lap tartalmát. Ez helyes is, nehogy tesztmenetből át tudjon menni a tanuló a tesztkészítő módba, és meg tudja nézni az egész tesztet írási üzemmódban, vagy akár bele is írhasson. A RESET hatásának emlegetése mindenesetre a szöveg későbbi részében is elég lett volna. (Mellesleg az online help is az elején emlegeti.)

Az online help elérése látszólag elég egyszerű. Sajnos azonban a hosszabb szövegek átnézése alatt a HELP gombot folyton lenyomva kell tartani, ami elég fárasztó. Miért nem lehetett valami ki-bekapcsolós megoldást találni? A help szöveg magyar nyelvű, mindig a szituációhoz illő, csak éppen, mint már említettem, magyartalan.

Még egy botránykő van a programban. Hol angolul, hol magyarul akar beszélni földhözragadt magyar ügyfelével. A legidegesítőbb a bevezető angol nyelvű kérdés, ami ráadásul nem is igazán lényegre törő. Szerintem azt kellett volna megkérdezni, hogy tesztmenetet (T) vagy tesztkészítést (K) akar-e az ügyfél.

A sok szidás ellenére a CHARLIE jó program, mint az a kevesebb szubjektivitással megkonstruált értékelő táblázatból is kiderül. Aki megtanulja a használatát, igen sokoldalúan, meghökkentően sokféle funkcióban fel tudja használni.

A program a használójának esztétikailag is kellemes élményt nyújt. Szép az indító kép, és ablakozási technikája is kiváló. Zenéje olykor ügyetlen, de a győzelmi himnusz csak növelheti a sikerélményt.

Van tehát egy újabb pedagógusi szerszámosládákba illő program, amelyet drillek és tesztek készítésére bárki más is használhat. Kvízjátékok szerkesztéséhez is jól hasznosítható, ezért a kultúrházakban és a szórakoztató iparban is odafigyelhetnek rá. Én végül is eléggé kiismertem CHARLIE-t, és bátran ajánlom mindenkinek.

Zsadányi Pál

ÖSSZESÍTŐ ADATOK

Forgalmazó:	*Novotrade*
Terméknév:	*CHARLIE*
Szerzők:	*Zawiasa Péter NYÚLSOFT és Sík Péter nyelvtanár*
Géptípus:	*Plus/4, Videoton TVC*
Hordozó:	*kazetta*
Dokumentáció:	*egy kazettaborítónyi méretű lap*
Ár:	*424 Ft*

MINŐSÍTŐ ADATOK

Kezelhetőség:	*jó*
Teljesség:	*közepes*
Dokumentáltható-ság:	*közepes*
Használhatóság:	*kiváló*
Ár/teljesítmény:	*kiváló*
Összbenyomás:	*jó*

ADOK—VESZEK—CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: kereskedelmi tevékenységet folytatóknak gépelt soronként (60 karakter) 100,- Ft, másoknak az első sor 50,- Ft, minden további sor 20,- Ft. Kérjük, hogy a hirdetés díját az NJSZT OTP V.ker. fiókjánál (218-98055) vezetett 508-8609 rendezvényszámlára vagy 1054 Bp., Báthori u. 16. címére fizessék be, rózsaszínű postautalványon (jelölve, hogy apróhirdetés), a befizetést igazoló szelvényt pedig csatolják a hirdetéshez. Hirdetéseiket a szerkesztőség címére várjuk (1371 Bp., Pf.: 433). Az NJSZT tagjai továbbra is kedvezményesen hirdethetnek (az első sor ingyenes), de kérjük, hogy adják meg tagsági számukat. Azokat a hirdetéseket, amelyek a hónap első napjáig megérkeznek, már a hónap végén viszontláthatják lapunkban.

ADOK

Amiga programok nagy választékban eladók! Érdeklődni lehet: Keresztes Gábor, Bp., Laky köz 11. 1142. Tel.: 1643-452

AMIGA !!! THE WINNER IN HUNGARY !!! Legújabb programok minden mennyiségben. Laci Szőnyi, Bp.,Tavirózsa 5. 1161. Tel.: 1848-471

Amiga és **C64** programok (külföldi forrásból) eladók! Minden program 1-2 hetes töréső. Érdeklődni lehet: (FBI CREW) C64: 06-1-1200772, Amiga:06-1-1224529

Amigára (100 Ft/disk), **C64-re** (50 Ft/lemezoldal) programokat adok! (A legújabb '89-eseket is!) **Amiga** cserepartnereket keresek! Eladók a Mikroszámítógép Magazin, Mikrovilág, C=Újság, Spectrum Világ 88-as évfolyamai! Eladó C64 + magnó + joystick + resetkapcsoló (18 000 Ft); Citizen 1200 + 2 festékkazetta + interfész (30 000 Ft); 30 db kazetta új C64-es programokkal (egy kazettán kb 40-50 prg., 1 db prg. 15 Ft) + szakkönyvek együtt vagy külön is! Csak komoly érdeklődők jelentkezzenek! Valent Gábor, Nyíregyháza, Északi krt. 21. 4/17. 4400

Amigások! Az AMIGA Magazin 5/88-tól 9/89-ig (17 db) + 1 db AMIGA Aktív komplett 6000 Ft-ért eladó. (Tart.: teljes assembler tanfolyam, assembler forráslisták, programleírások, C nyelv stb.). Programcsere is! HIGH VOLTAGE Balassagyarmat, Pf.: 118. 2661

Atari 130 XE (128 k RAM) + datasett eladó. Minden XL/XE jelű Atarival kompatibilis (pl. 800 XL). Árajánlatot a következő címre kérek: Balla Tibor, Tatabánya II., Erdész u. 26. 2800

C Plus/4, magnó, 1551 floppy, 2 db joystick, 30 db lemez, 150 program, szakkönyvek eladók. Irányár: 35 000 Ft. Orosz István, Dunakeszi, Keleti u. 18. 2120

C16, C Plus/4, C64, C128, Amiga 500 gépekre a legújabb 1989-es programok eladók. Keresztfalvi János, Budapest, Doberdó út 4. 1034. Kérésre listát küldök!

C16-os és **C Plus/4-es** programok olcsón (8 Ft/db) eladók. Válaszborítékért listát küldök. Lajos Róbert, Szeged, Szilléri sgt. 24/A. II/6. 6723

C16, C Plus/4 programok, köztük a legújabb '89-esek eladók - 10 Ft/db - kazettán. Listát kérésre küldök. Agárdi Tibor, Kecskemét, Dankó P. u. 37. 6000

C Plus/4 + magnó + 2 joystick + sok program + szakirodalom eladó. Tájékoztatót küldök. Szuperprogramok - szuperáron - külön is! Regős Attila, Gara, Kossuth u. 38. 6522

C64 floppyval, 70 db lemezzel, 2 joystickkal 40 000 Ft-ért - külön kazettás programok kazettánként 360 Ft-ért - eladók. 1500 programról listát küldök. Tóth Kornél, Nagykálló, Ady út 28. 4320

Eladó: **C64** + 1541-es meghajtó + magnó + Junoszty TV + programok lemezen és kazettán. Pusztafalvi Szilárd, Pécs, Jurisics M. u. 1. 7624. Tel.: 24-054

Commodore 64 magnóval és programokkal olcsón (15 000 Ft) eladó. Cím: Steingart Ferenc, Budapest, Budaörsi út 95-101. 1118

C64 alapgép + Speed Dos + Kernelek (16.000,- Ft); C1541 (régi) + Speed Dos + 40 Track (26.000,- Ft); SP 180 Seikosha nyomtató (20.000,- Ft) 1351-es egér (5.000,- Ft) és gyorstöltő modem (2.000,- Ft) eladó. Bojtor Tibor, Zalaegerszeg, Átalszegett út 153. 8900. Tel.: 11-373

C64 II. számítógép és VC 1541 floppymeghajtó kedvező áron eladó. Leszkó József, Vác, Király u. 10. I/3. 2600

C64-es lemezek (400 Ft/doboz) eladók. Szabó Géza, Nyíregyháza, Városmajor u. 26. fsz. 1. 4400

Commodore 64-hez 64 k-s cartridge felhasználói programokkal eladó. Tel.: 166-9292 este

C64-hez több mint száz magyar ékezetes karakterkészlettel, nagy grafikagyűjteménnyel és magyar nyelvű leírással eladnám PAGEFOX-omat. Sallay Zsolt, Székesfehérvár, Ybl u. 2/c. 8000

Eladó C64 beépített reset kapcsolóval (18 eFt), 1541 drive + kb. 70 db lemez programokkal 100 db-os tárolóban (25 eFt), datasette + kb. 14 db C-60 kazetta programokkal (5 eFt), C64-es szakirodalom 4500 Ft értékben (2 eFt), 250 db kibontatlan lemez (50 Ft/db). **Együtt az egész 10 %-kal olcsóbban! Keresek Amigához** magyar nyelvű hardver- és szoftverleírásokat. Pajer Ferenc, Budapest, Tóvirág u. 2. VI/26. 1108

C64-es számítógép + magnó, 2 joystick sürgősen eladó! Bp. XXI. Vas Gereben u. 15. (4 óra után)

C64 programok eladók kazettán és lemezen (10-15 Ft/db), cserélek is. Listát válaszborítékban küldök. Nyitrai József, Dunaújváros, Szabadság út 4. fsz. 2. 2400

C64-re programokat eladok kazettán. Kétezer programom van! Egy db ára 4 Ft. (Diskes program is kazettán! Tel.: 26/45-106. Pintér Balázs, Érd, Petőfi S. út 113. 2030

C64-es lemezek eladók (200 db) 100 Ft/db szuper programokkal. Kérésre listát küldök. 1351-es egér, Citizen 120D + interfész, magnó (6000, 25 000, 2000 Ft) eladók. Németi Ferenc, Budapest, Nagyenyed u. 8/a. 1182

C64/128-hoz RS232 interfész és Centronics nyomtatóinterfész, valamint PC/AT-hez tápegység eladó. Kertész Zoltán, Budapest, Gőzmozdony u. 6. 1108. Tel.: 177-8171 este

C128D olcsón eladó! Tel.: 06/76/61-116

Commodore 128 számítógép + joystick (irányár 19 500 Ft), Ferguson 80 oszlopos monochrome zöld monitor (9500 Ft), Citizen IDP-560 kétszínű nyomtató (7500 Ft), zárható lemeztartó - 100 db-os (1000 Ft), 60 db floppylemez (C64-es, C128-as, CP/M-es programokkal 95 Ft/lemez) eladó! Érdeklődni lehet: Prech Zsolt, Tatabánya, Ifjúmunkás u. 21. IV/3. 2800

Szeretnék kapcsolatba kerülni **Commodore 64 személyi számítógépek tulajdonosaival** programcsere, programmásolás céljából. Commodore 64 gépre játékprogramokat, rajz-grafikai programokat, zenei programokat másolásra bocsájtok, magánszemélyek részére. 1 db játékprogram ára: 400 Ft; 1 db rajzprogram ára 200 Ft; 1 db zenei program ára 200 Ft. Keresek C64-re korszerű grafikai programot. Czepek Zoltán, Noszlop, Sport ut. 50. 8456. Érdeklődni lehet 16 óra után.

Commodore 116 belső bővítővel (= Plus/4) sürgősen eladó (8000 Ft)! Éliás Sándor, Székesfehérvár, Schönherz Z. u. 8. 107. szoba. 8000. Tel.: 22/16-377 (H - P 18 h után)

Commodore 128D, Philips monitor és szoftverek olcsón (!) eladók. Töreky Sándor, Budapest, Budaörsi út 95-101. B ép. 107 sz. 1118. Tel.: 166-7788 (21 óra után)

Eladó Enterprise 128 + magnó + joystick + exdos controller + floppy (720 k, 32 k/sec). + monitorkábel + programok + 20 lemez (3M), kazetták + szakirodalom + játékleírások. Ár: 45 000 Ft. Kása József, Pécs, Magyarürögi u. 68. 7634. Tel.: 72/35-400 (hétvégén)

Enterprise programokat adok, veszek, cserélek. Spectrum programok olcsón eladók. Sváb László, Budapest, Pára u. 8. 1108. Tel.: 177-4243

Enterprise programokat adok, veszek, cserélek. Közel hatszáz programmal rendelkezem. Lemezre (5.25-ös) is másolok. Keresem a gépen futó CP/M programokat. Jelszavam: jó minőség, gyorsaság, legolcsóbb árak. Válaszborítékért listát küldök. Szabó József, Bonyhád, Bezerédi u. 41. III/5. 7150

Enterprise programokat adok és cserélek. Több mint ötszáz program a legújabb átiratokkal. Jó minőség, olcsó árak. IBM formátumú lemezen is! Válaszborítékért listát küldök. Barabás Barbara, Bonyhád III. Pf.: 25. 7153

Enterprise (15 000) és Primo A64 számítógép (7500) forintért eladó. Torkos Csaba, Várpalota, Steinmetz u. 20. 8100

Enterprise-hoz lemezmeghajtó (MOM, $5^1/_4$-es, 2x180 kbájt, 2x40 track) rendszerlemezzel 4000-ért eladó. Horváth Zoltán, Bp., Zöldfa u. 11. 1214

Enterprise programok adok és cserélek. Válaszboríték ellenében listát küldök. Olcsó ár! Mészáros Gergő, Fűzfőgyártelep, Nike krt. 17. 8184

Enterprise programok eladók (10-50 Ft). Válaszborítékért listát küldök. Lesné, Budapest, Delej u. 51. XV.1h. IV.em. 1089

ITT számítógép CP/M operációs rendszer alatt eladó monitorral és 7 MB külső merevlemez + beépített floppy drive, esetleg elcserélhető színes TV-re vagy videomagnóra. Kretz Ádám, Békéscsaba, Kulich tér 1. III.14.

Primo A64-es szakkönyvekkel eladó. Irányár: 5000 Ft. Lénárt Barnabás, Tel.: 140-6566

Seikosha GP50 printer Spectrumhoz eladó. Keresek Amiga szótárprogramot. Gulyás Ákos, Bp., Endrődi 42/D. 1026. Tel.: 136--7543

Spectrumra mindig a legújabb programok eladók! Ingyenes katalógus! Pilláry Gábor, Pécs, Bajcsy-Zs. u. 4. 7622

Spectrum 48/128 programok olcsón, nagy választékban, garanciával kaphatók. Válaszborítékért részletes katalógust küldök. Minden megrendelő ajándékot kap! Kiss Gábor, Győr, Venyige u. 31. 9028

ZX-Spectrum felhasználói és játékprogramok eladók. Válaszborítékért katalógust küldök. Bogyó Frigyes, Pécs, Bihari J. u. 3/B. fsz.1. 7633

ZX-Spectrum (48 k) + interfész + joystick + beep erősítő + több száz program olcsón eladó. Szilágyi József, Budapest, Párkány u. 28. 10/60. 1138

Adok hibás /ULA, PROM/ **ZX-Spectrum** 48 k-t trafóval. Árajánlatot a következő címre kérek: Faludi Olivér, Balassagyarmat, Rákóczi fejedelem útja 87. 2660. Tel.: Balassagyarmat 83-93

ZX-Spectrum (48 k) + magnó + programok + irodalom + joystick + fényceruza ! mikropoker olcsón eladó. Érdeklődni lehet 16 óra után. Bicsák Lajos, Bp., Agyagfejtő u. 18. VII. 29. 1108

ZX-Spectrum színvonalas programokat adunk-cserélünk. Minden 10. program után 1 ingyenprogram! Sándor/László Company, Devecser, Gábor Á. u. 1. 8460

Eladásra kínálom nagy mennyiségű **Sinclair ZX81** programom és szakirodalmam. Esetleg csere is érdekel. Menyhárt Tibor, Debrecen, Gyepűsor 12. 4031

ZX-Spectrum 128 k (keveset használt, jó állapotban lévő számítógép), magnóval, 55 db kazettával (600 program), Interfész 1-gyel, Mikrodrive-val, 5 db Mikrodrive kazettával (cartridge), nagyon sok irodalommal, interfész 2-vel, 2 db joystickkal, Timex 2040 nyomtatóval + 9 tekercs papírral sürgősen eladó 56 000 Ft-ért! Érdeklődni lehet: Kiss Henrik, Budapest, Határ utca 103. 1213

TV-Computer programokat adok-veszek-cserélek. Programkatalógust kérek. Krajnyák Levente, Győr, Munkásőr u. 40.9023. Tel.: 06/96/28-159

TV-Computer programokat adok-veszek-cserélek. Listát, válaszborítékot kérek! /Közel ezer program.) Molnár János, Szolnok, Jászi F. u. 10. VI/25. 5000. Tel.: 56/31-085

VC 1541/II. floppy és 10 db lemez (külön is) olcsón eladó. Tel.: 177-9049

A bosszú, Új vadnyugat és az **Új vadnyugat II.** programok - magyar nyelvű illusztrált kalandjátékok C64-re - megrendelhetők utánvéttel: Rátkai István, Esztergom, Bocskoroskuti út 28. 2500. Áruk lemezen 370, kazettán 340 Ft. (A mellékelt leírás és a hordozó árával együtt!)

Centronics nyomtató eladó. (Seikosha SP-1000I). NLQ, 128 írásmód. Ár: 20 000 Ft. Faragó Zsolt, Veszprém, Klapka u. 4/C. 8200

Fordításokat készítek a **64'er Magazinból**. Több mint 3000 oldalnyi kész anyag! Kedvező árak. Textomat+, GEOS 1.2, Hi-Eddi, Giga CAD, Disc Wizard, Disc Demon stb. dokumentációk. Szolnoki Béla, Bp., Pf.: 400. 1446. Válaszborítékot kérünk!

Joystick Szerviz a Flórián Áruházhoz közel! Javítás, Magnófej-beállítás. C64 játékprogramok kazettán és floppyn 10 Ft/db. Bp. III. Kerék u. 36. IV/24. Hétfőn és szerdán 17-től 19-ig.

Jutányos feltételekkel vállalaok középfokú német nyelvvizsgára való fölkészítést. C64 vagy C128 és floppy szükséges. Érdeklődni: Szolnoki Béla, Bp., Pf.: 400. 1446

Karácsonyi ajánlatunk:
1 db fényceruza + 1 db joystick + program C64-hez vagy Plus/4-hez, postai utánvéttel most csak 1550 Ft! A karácsonyfa alá kerülhet, ha december 15-ig megrendeli. Címünk: CIMPUTEAM Kaposvár, 7400

MOM MF 6400, MOM MF 3200 Olivetti HT 090-es 8" lemezmeghajtók eladók. Nagy Jenő, Győr, Táko u. 2/A. 9025

NA VÉGRE! Minden együtt! 20-szoros turbo, RESET és programtörő (FREEZE) gomb, szövegszerkesztő, monitor, nyomtatóinterfész, óra, számítógép, kibővített BASIC és mindez egy cartridge-ben C64-hez, C128-hoz. Kívánságra bővebb tájékoztatót küldünk vagy bemutatjuk használatát. Tel.: 184-8845

Üres 20 márkájú ($5^1/_4$-es lemezek eladók. 70 Ft/db. Válaszborítékot kérek. Commodore 64, floppyval 2 joystickkal, lemezekkel és szakkönyvekkel eladó. Farkas Zsolt, Kalocsa, Magyar L. u. 21. 6300

Vegyen programokat a **vip-től!** Legfrissebb **importok C64** és **Amiga 500** gépeken! Áraink: Amiga: 70 Ft/disk; C64: lemezen 40 Ft/disk, kazettán 10 Ft/prg. Sonnevend Lőrinc, Bp., Bartók Béla út 8. I/4. 1111. Greetings to: ADT, TRI, ILC, TGS, GENIVS, Trays, FBI!

8 MHZ-es XT 640 k RAM-mal, 1,2 MB-os floppyval, sárga 14 collos monitorral, MCGP kártyával, 83 gombos billentyűzettel, valamint 2 db 12 MHZ-es AT alaplap megegyezés szerinti áron eladó. Sárosi Sándor, Környe, Zsíros út 1/A. Tel.: 34/72-044

5,25" DS-DD lemezek korlátlan mennyiségben eladók. (89 forint 1 db). Sasvári Gábor, Lenti, Petőfi út 33. 8960

VESZEK

C64-re keresem a Dun Darach, a Marsport, a Neverending Story és bármilyen szintű tarokk programot. Ezenkívül a listát küldőknek válaszolok. Wersényi György, Győr, Vajcsuk L. u. 31. 9024

Enterprise EX-DOS lemezvezérlőt vásárolnék. Hajagos Péter, Veszprém, Bakony u. 18. 8200. Tel.: 80/22-802

Műszaki egyetemistának vizsgamunkához szüksége lenne egy hibás, esetleg totálkáros C64-re! Tel.: 1128-349

CSERÉLEK

C16-ot magnóval elcserélek C64-hez való floppyra, vagy eladó C64-es programokat cserélek. C16, Plus/4-es programok eladók. Listát kérek és küldök. Tóth József, Pápa, Pf.: 22. 8501

C16, Plus/4,C64-es programokat cserélek és eladok lemezen és kazettán. Lékó Pál, Szolnok, dr. Csanádi krt. 8. VI/18. 5000

C64-es programokat cserélek vagy olcsón eladok kazettán. Listát kérek és adok. Csontos Attila, Szeged, Dorogi Imre u. 7. 6753

Commodore 64-es programokat cserélek lemezen és kazettán. Hajdu László, Budapest, Szigeti J. u. 17. V/32. 1041

C64 és **Spectrum 48** programokat cserélek, adok és veszek. Cserénél listát kérek. Vételnél válaszborítékért listát küldök. Keresem C64-re a Hillsfar és a Pool of Radince kódjait és a Panzer Stike programot. Spectrumra keresem a LORD OF THE RINGS és LASER SQUAD programot. Lendvai Tibor, Kazincbarcika, Szegfű u. 13. 3700. Tel.: 48/12-715

C64-re programokat cserélek lemezen. Vennék BARD'S TALE 3.-leírást, de más leírások is érdekelnek. Listát kérek. ifj. Horváth Lóránt, Érd, V. ker. Arany J. u. 43. 2030.

C64-re, kazettán, lemezen 2200 programból cserélek, eladok (10-20 Ft). Németh András, Győrújbarát, Veres P. u. 23. 9081

C16, C Plus/4 és C64-es programjaimat kínálom fel cserére, szalagon és lemezen! Főleg oktató- és játékprogramok érdekelnek, lehetőleg listát kérek! Várom a régi cserepartnereim jelentkezését is! Hika György, Szeged, Sárosi u. 1/A. 6724. Tel.: 62/30-496 (17 óra után!) Csak cserélek!

Enterprise programokért cserébe Enterprise és Spectrum programokat adok. Listát kérek! Szarka Endre, Pápa, Mikes K. u. 11. 8500

Enterprise programcsere + eladó egy SPEAK EASY. Maszlin Zoltán, B.földvár, B.szentgyörgyi út 62. 8623

Seikosha gp-250X típusú nyomtató EPROM-ját vagy annak tartalmát tetszőleges adathordozón megegyezés szerint elcserélem valamire. MPS 803-as nyomtató, 64 k-ra bővített PRIMO B32-es és $5^1/_4$-es MOM floppy eladó. Lőrentei Tamás, Veszprém, Diósi M. u. 2/d. 8200

Spectrum 48 k programokat cserélek. Listát kérek. Cserealapom kb. 900 program. Baksa Béla, Bp., Bányász u. 12. 1205

Színvonalas (87-89-es) programokat cserélek **ZX-Spectrumra**. Leveleket listával kérek a következő címre: Bodnár Péter, Miskolc, Engels út 33. V/3. 3529

TVC programokat cserélek! Listát kérek! Árvai László, Miskolc, Középszer út 90. 5/2. 3529

3 1/2"-os diszket adok 3" (Schneider)-ért. Esztergár Pál, Bp., Fehérvári út 35. 1117. Tel.: 161-0171

Coffron, J. W. — Long, W. E.: Mikroprocesszoros rendszerek illesztési technikája (Budapest, 1989. Műszaki Könyvkiadó, 338 oldal. Ára: 280 Ft.)

A könyv azoknak a berendezéseknek, illetve áramköröknek a már meglévő mini-, mikrovagy személyi számítógépes rendszerekhez való illesztéséről szól, amelyek segítségével a szóban forgó rendszerek kapacitását megnövelhetjük, szolgáltatásaikat bővíthetjük. Középpontjában mindig olyan számítógépes rendszer áll, amelynek kihasználása az adott esetben elérte az eredetileg tervezett határt, vagy amelynek nincsenek meg a felhasználója által igényelt képességei.

A kötet az olyan mérnököknek, technikusoknak íródott, akik talán először tanulmányozzák az illesztéstechnikát, vagy akik gyakorlati útmutatást és ötleteket keresnek.

A különböző fejezetekben tárgyalt témakörök között megtalálható az illesztés ROM-hoz, a statikus és dinamikus RAM-hoz, a különféle beviteli-kiviteli portarchitektúrákhoz, valamint az ECVL és a CMOS logikai áramkörcsaládokhoz, a multiplex üzemű hétszegmenses kijelző vezérlése, az analóg-digitális és digitális-analóg átalakítás, egy programozható beviteli-kiviteli integrált áramkör alkalmazása, továbbá az S-100 sínrendszer, a soros-párhuzamos adatátvitel, az elektronsugárcsöves megjelenítő, a billentyűzet illesztése és hasonlók. Egy teljes fejezet egy népszerű iskolaszámítógép, a TRS-80 illesztéséről szól.

100+4/3. Játékok és felhasználói programok C16-PLUS/4, Aliens, Storm, Tir Na Nog, Tedpaint, Easy Script, Sys Cracker (Budapest, 1989. LSI ATSZ, 110 oldal. Ára: 137 Ft.)

A népszerű játékprogram-sorozat 3. kötetének tartalma az olvasók kívánságára a korábbi kötetekétől némileg eltér. A szerzők felsorolják az összes rendelkezésükre álló — és eddig még meg nem jelent — POKE-okat. A mikrolexikonból kihagyták az iskolai, illetve az oktatóprogramokat. A játékprogramok közötti könnyebb eligazodást segíti a játékok osztályozása (értékelése).

A játékleírásoknál a bonyolultabb programokat részletesen ismertetik, és a Programozástechnika című fejezetben segítséget nyújtanak néhány nehezen megoldható feladat teljesítéséhez is.

Barakonyi Károly: A Framework II használata kezdőknek (Budapest, 1989. LSI ATSZ, 297 oldal. Ára: 343 Ft.)

A 80-as évek a számítástechnika alkalmazásában gyökeres fordulatot hoztak. A hardver és a szoftver fejlődése megnyitotta az utat a tömeges méretű alkalmazások elterjedése előtt: olyan felhasználók is kapcsolatba kerülhettek a számítástechnikával, akik klasszikus értelemben vett számítástechnikai előképzettséggel nem rendelkeztek.

A fejlesztés eredményeként pedig megjelentek és elterjedtek a barátságos, integrált jellegű rendszerek. A Framework II ennek a fejlesztési hullámnak az egyik legjelesebb képviselője.

A Framework II olyan integrált programcsomag, amelynek segítségével összetett dokumentációk részei (szöveg, táblázat, adatállomány, ábra), valamint a dokumentáció egésze (vázlat, szerkezet, folyamatos nyomtatás) egyaránt jól és kényelmesen kezelhető. A rendszer a kereteken (frame) alapuló új szoftvertechnológiát használja fel. Ennek eredményeként a feladat megoldása elemeire bontható, az elemi feladatok önállóan kidolgozhatók, majd az építőkockákból összerakható a feladat egésze. A részek viszonylagos önállósága és egyszerű kapcsolhatósága lehetővé teszi, hogy a természetes emberi logika szerint közelítsünk a megoldandó feladathoz. A Framework II fejlesztői rugalmas, áttekinthető és könnyen kezelhető eszközt adnak a felhasználók kezébe.

A könyv elsősorban a számítástechnikailag képzetlen felhasználóknak íródott, minimális ismereteket feltételez az IBM PC-kkel és az operációs rendszerekkel kapcsolatban.

Dr. Poronyi Gábor (szerk.) Számítástechnikai versenyfeladatok II. (Pécs, 1989. Baranya Megyei Pedagógiai Intézet, 82 oldal. Ára: 90 Ft.)

A kötetben a Baranya megyei számítástechnikai tanulmányi verseny (1988—1989), a Bács-Kiskun megyei általános iskolai számítástechnikai tanulmányi verseny (1987—1989), a Pest megyei számítástechnikai verseny általános iskolásoknak (1988), a Szeged városi „MIKROMAT" számítástechnikai verseny (1987), valamint a Tata városi számítástechnikai verseny (1988) feladatait, azok megoldásait és értékelési szempontjait találhatjuk meg.

Megrendelhető a kiadónál: 7621 Pécs, Széchenyi tér 3.

Bánné Varga Gabriella: Programtervezési gyakorlatok dBASE kódokkal COBOL, PL/1, Pascal, FORTRAN kódolási ajánlásokkal Jackson-módszer szerint (Budapest, 1989. SZÁMALK, 191 oldal. Ára: 308 Ft.)

A 60-as években, amikor már mind nagyobb és bonyolultabb feladatokat bíztak a számítógépekre, bekövetkezett a szoftvergyártásban a röviden csak „szoftverkrízisnek" nevezett jelenség. A működésre átadott programok az üzemelés során katasztrofális hibákat produkáltak, a termékek gyorsan elavultak, és keményen ellenálltak a módosításoknak. A magyarázat részben a problémák bonyolultságában, részben a szoftverkészítés manufakturális jellegében volt keresendő. A bajok orvoslására született meg először a strukturált programozás. Ez a módszer nagy horderejűnek látszott, de hamar kiderültek a hiányosságai is. A szoftverkészítési folyamat átalakítása nem állt meg a programozásnál, hanem az ipari termékek gyártási folyamatait figyelembe véve kezdték az adott terület technológiai lépéseit is mind jobban felismerni, a különböző fázisokat módszerekkel megtölteni. A strukturált programozás témaköre is kibővült, azt továbbvivő programtervezési módszertanok nyertek alkalmazást (például Warnier módszere, a Jackson-féle programtervezési módszer). Ekkor robbant be a számítógépek világába a személyi számítógép.

A szerző az oktatási tapasztalatok alapján állította össze a programtervezési feladatokat, figyelembe véve a fokozatosság elvét. A kötet mellékleteiben kódolási ajánlások találhatók a legelterjedtebb programozási nyelvekre (COBOL, PL/1, Pascal, FORTRAN), néhány feladatnál dBASE nyelvű implementációt is közöl.

Kalózkongresszus

Amszterdamban világkongresszust tartottak a számítógépes kalózok, azaz az olyan műkedvelők, akik hobbiból gépük segítségével behatolnak a számítógépes hálózatok adatbázisaiba. A háromnapos tanácskozáson a mintegy kétszáz résztvevő kifejtette, hogy nem tartja magát bűnözőnek, ellenkezőleg, a hivatalokat vádolták meg hanyagsággal, mondván, hogy azok nem tesznek meg mindent az adatok védelme érdekében.

Zsebszámítógép

Az Intel a közelmúltban dobta piacra a Wilcard 88 elnevezésű rendszerét. Ez egy 6x12 centiméteres lapon mindazokat az elemeket tartalmazza, amelyek az IBM PC/XT mikroszámítógéppel egyenértékű működést tesznek lehetővé. A lapon elhelyezett mikroprocesszor az energiatakarékos Intel 80C88 típus, ez a mikroszámítógépekkel azonos feladatok elvégzésére képes. A kártyán egyéb áramköri egységek is elhelyezkednek, közöttük azok, amelyek a memóriával való együttműködést biztosítják. Magát az operatív tárat a lap nem tartalmazza, de a lehetőség adott az elhelyezésére. Ugyanitt helyezhetők el azok a kereskedelmi forgalomban levő kártyák, amelyekkel az eszköz a legkülönbözőbb feladatok elvégzésére alkalmassá tehető. A rendszer nem alkalmas programbelövésre, hanem kifejezetten a kész programok futtatásáig van lekopaszítva. Példa lehet erre a gépkocsi, amelyben a kifogástalan működéshez szükséges valamennyi adatot — a futófelület állapotától a belső tér klimatikus viszonyaiig — folyamatosan ellenőriz a rendszer, működés közben valamennyi funkciót egymással összhangban vezérli és optimálja, az üzemeltetési költségeket kiszámítja, és a vezető számára szükséges információt a műszerfalat helyettesítő grafikus kijelzőn mutatja. Mindezt egy célorientált szabványos szoftver segítségével végzi.

Lemezszerviz

2,5—300 Mbájtos mágneslemezcsomagok felújítására és forgalmazására szakosodott az idén alakult Kürt Kft. Vállalják winchester-tárak szervizét is, mégpedig a gyártóktól beszerzett új alkatrészek felhasználásával. A kft. kialakította az alkatrészek cseréjéhez szükséges, a gyártók által előírt tisztasági körülményeket is.

Négy megabit

A világ második legnagyobb áramkörgyártó vállalata, a japán Toshiba bejelentette, hogy 4 Mbit kapacitású EEPROM-ot fejlesztett ki. Az EEPROM tulajdonsága, hogy adatai kitörölhetők és újraírhatók, ugyanakkor az adatokat akkor is biztonságosan megőrzi, ha a tápenergiáját kikapcsolják: azaz ötvözi a DRAM-ok könnyű újraírhatóságát és az EPROM-ok kikapcsolás utáni tárolási tulajdonságát. A jövőjét tekintve várhatóan a sok mozgó alkatrészt tartalmazó winchester-tárakat válthatja fel. Míg a DRAM-ok körében ma már nem számít újdonságnak a 4 Mbites tároló, az EEPROM-ok között még a 256 kbites tekinthető elterjedtnek.

IBM-fejlesztések

Óriási összegeket fordít az IBM az új PS/2 család bevezetésére. Csupán az egységes felhasználói környezetet biztosító SAA (System Application Architecture) rendszer kidolgozása és elterjesztése várhatóan több milliárd dollárjába kerül a cégnek. Az SAA valószínűleg döntő hatással lesz a számítástechnika jövőjére, alapvetően meghatározza majd a szoftverfejlesztések irányát.

Hasonlóan jelentős lépés a hardver területén az IBM mikrocsatorna, az MCA (Micro Channel Architecture). Bár csak 1988. június 2-án mutatta be az IBM a hét új asztali számítógépét, melyekben már az MCA-technológiát használták, már 1988 végéig több mint 600 MCA-bővítőkártya került forgalomba, különféle célokra. A mikrocsatorna nem egyszerűen új, bővített kártya-összekapcsolási séma, hanem olyan rendszer, amelyben az egyes kártyákhoz programozható címek, megszakítási és prioritási szintek rendelhetők. A konfigurálás nem átkötésekkel vagy kapcsolókkal, hanem a POS (Programmable Option Select) szoftver segítségével történik. Az MCA várhatóan a közeljövőben gyakorlatilag szabvánnyá válik, de máris megjelentek azok az illesztőegységek, amelyek elterjedt rendszerekhez biztosítják a csatlakozást.

Jóslat

Az amerikai Computer Technology Research cég a közelmúltban részletes tanulmányt publikált a személyi számítástechnika jövőbeli tendenciáiról, ezen belül elsősorban az IBM fejlesztéseinek előrelátható hatásáról. A tanulmány szerint 1989-ben az IBM szoftverforgalmának várható megoszlása: DOS = 91%, OS/2 = 9%. Ugyanez az arány 1992-ben a becslések szerint: OS/2 = 62%, DOS = 38%.

Választások

Egyre intenzívebben folynak a választások technikai előkészületei. Az előzőek idején a megyékből szirénás futárszolgálat szállította a Parlamenthez a jegyzőkönyveket. Jelenleg kétlépcsős számítógépes kapcsolatot építenek ki a megyék és a Parlament között. Ennek segítségével a korábbinak mintegy felére, 12 órára csökken az az idő, amely az urnák lezárásától az előzetes eredmények kihirdetéséig eltelik.

A számítógépek a gyorsaságon túl a választási eredmények megbízhatóságát, nyilvánosságát és ellenőrizhetőségét is szolgálják. A parlamenti központban a folyamatos tájékoztatást elektronikus megjelenítés könnyíti meg: jókora tábláról olvasható majd le — az urnák lezárását követően — a kétóránkénti helyzetkép. A számítógép igény szerint csoportosítja az eredményeket, kiírja például, hogy az egyes pártok az adott időpontban hány mandátumhoz jutottak.

Az elgondolások szerint a 350 országgyűlési választókerület egy részére is kiterjed a számítógépesítés, elsősorban azokban a városokban, amelyek az állami népesség-nyilvántartás alközpontjai. A kiépített technika a választásokon kívüli időben a közigazgatás munkáját segíti.

Kiadói rendszerek

Az idén alakul Typoprint DTP-Centrum a nyomdai előkészítést segítő szoftvercsalád készítésén munkálkodik. A technológia legelején a Typo-color szoftver áll, amely színes képolvasók illesztését végzi a nyomdai előkészítő rendszerhez, beleértve a beolvasott képek nyomásra való előkészítését is. A beolvasott képek rácsra bontását a Typo-raszter végzi. A hazánkban is rendkívül elterjedt LaserJet II-höz PostScript jellegű lehetőségeket biztosít a Typo-script nevű program. A komoly érdeklődéssel várt Typo-család várhatóan 1990 januárjában jelenik meg.

Informatikai iskola

Érdekes kísérletet kezdtek Budatétényben: az általános iskola negyedik osztálya után kezdik a nyolcosztályos középiskolát. Ebben elsősorban az informatikai képzés lesz az, ami más. Az induláshoz 26 darab IBM PC-vel kompatibilis géppel felszerelt laboratórium áll az iskola rendelkezésére. Ezt a gépparkot a közeljövőben ötvenre kívánják felfejleszteni. Az iskola elvégzése a diákok számára középfokú számítástechnikai képzettséget is nyújt.

City Taxi

Az 1983-ban alakult City Taxinál jelenleg 900 fő dolgozik, és havonta 100—110 ezer megrendelést vesznek fel. Jelentős fejlesztési eredményük, hogy a megrendelések fele már számítógépen érkezik. Állandó megrendelőik — vendéglátóhelyek, szállodák, egészségügyi intézmények — egy kódszámot kapnak. Ennek birtokában felhívják a City Taxi számítógépes telefonszámát, bemondják a kódszámot, például 215, és leteszik a kagylót. A City Taxi operátora ezt a számot begépeli a számítógépbe, és a képernyőn megjelenik: hova kell menni a kocsival. Ez rendkívül meggyorsítja a munkát: a taxirendelés átlagos időtartama 27 másodpercről 7 másodpercre csökken, mérsékli a telefonvonal foglaltságát, s nem utolsósorban megbízhatóbbá, pontosabbá vált a taxirendelés.

A City Taxi személyzeti nyilvántartása is számítógépen van. Így pillanatok alatt bárki kikereshető név, rendszám, hívószám, lakcím, telefonszám vagy akár nyelvismeret szerint is. Az utóbbi például akkor hasznosítható jól, ha betelefonálnak, hogy keresnek tíz angolul beszélő vezetőt: kikeresik mondjuk mind az ötvenet. S akkor ezeket a kollégákat megkeresve egyeztetik az időpontokat, s ha vállalják, mehet is a munka.

Finommechanikai és Elektronikus Műszergyártó Szövetkezet

1222 Budapest, Nagytétényi út 100-102. **1775 Bp. Pf.: 69.**
Telefon: 173-0011 **Telex: 22-60-34** **Telefax: 386-593**

Rendkívüli ajánlatunk

GP—01M típusú GraphyPlot berendezés nagyméretű, max. A;-es formátumú rajzok készítésére szolgáló, digitálisan vezérelhető rajzgép.
320 000,—Ft helyett 260 00,—Ft + ÁFA

GT—01 típusú GraphiTab asztali digitalizáló készülék, amely grafikus vagy képi információt számítógépes feldolgozásra alkalmas formába alakít.
90 000,—Ft + ÁFA

AMEDDIG A KÉSZLET TART!

További részletes felvilágosítással a Szövetkezet Kereskedelmi osztálya szolgál

Az Olvasó írja

Kovács Gábor és Jánoki Csaba, Budapest

Nagyon örvendetesnek tartjuk, hogy az Amigáról — amelyet a jövő házi számítógépének tartunk — végre aránylag színvonalas cikksorozatot indítottak meg. Véleményünk szerint az önök lapja már egy jó ideje leragadt olyan, régen elavult számítógépeknél, mint a C64, a Plus/4, a VIC-20, a ZX-Spectrum. Talán ennek is köszönheti csökkenő népszerűségét. Az Amigáról jelenleg semmilyen szakirodalom nem szerezhető be hazánkban. Ezért a konkrét leírásokat, adatokat, dokumentációkat tartjuk a legfontosabbaknak.

Néhány szót a lap borítójáról. A legnagyobb jóindulattal sem lehet figyelemfelkeltőnek vagy dekoratívnak mondani. Hiányoznak róla a tartalmat tükröző feliratok. A legtöbb számra igaz, hogy ha valaki nem olvassa el az újság nevét, rá nem jön arra, hogy ez egy számítógépújság.

Az Amigával kapcsolatos — most nem közölt — észrevételeiket eljuttattuk a sorozat szerkesztőjéhez, s bízunk abban, hogy a további cikkek — a növekvő terjedelemnek köszönhetően is — mindenki megelégedésére közölnek hasznos információkat ehhez a géphez.

Ami a Magazin csökkenő népszerűségét s az ennek okaként megjelölt géptípusokkal való foglalkozást illeti, engedtessék meg vitába szállnom. Kétségtelen, hogy ránk is hatással volt — bár ennek mértéke elenyésző — az az információrobbanás, amely az elmúlt egy év alatt következett be. A minimális példányszámcsökkenést mi nem a konkurens szaksajtó „elszívó" hatásának számlájára írjuk, hanem a fizetőképes kereslet csökkenésének tulajdonítjuk. Az embereknek most arról kell dönteniük, melyik kedves lapjukról mondjanak le egy-egy újabb kedvéért. Azt nem tudjuk lemérni, hányan olvassák könyvtárban a Magazint, mert kell a pénz a — 168 óra — c. lapra . . . Az elavultnak tekintett gépek pedig igenis a piacon vannak még, kötelességünk foglalkozni velük.

Végül a harmadik téma: a borító. Szerencsés — és egyben szerencsétlen — helyzetben vagyok, mert azt mondhatom, jelentős formai átalakuláson megy át a Magazin, már a következő számtól kezdve. Anyagi okokból el kell hagynunk a drága színes borítót, ennek „fejében" viszont éppen az önök által hiányolt tartalmi információk kerülnek majd a fekete-fehér címlapra. A lap ára is változik — 1 forinttal emelkedik — a kedvezőbb postai terjesztési besorolást megcélozva.

Ifj. Fekete László, Budapest

Mélységesen elítélem azokat, akik másolt programjaikat pénzért árusítják, és ezt az újságban hirdetik. Az egyik barátom írt egy, az „Adok — veszek — cserélek" rovatból kiböngészett ifjú legénynek, programcsere ügyében. Küldött is neki egy 60 perces Sony kazettát, programokkal. A legényke — látván a sok jó programocskát a kazettán — fogta, és zsebre vágta. A barátom 5—6 hónapja már jó pár figyelmeztető levelet adott fel számára, de választ még nem kapott. Mit lehet ilyenkor tenni? — kérdi ő. Mégsem utazhatja át a fél országot egy kazetta miatt! Vagy talán ezt kellene tennie? Nem tudom, minek hirdetnek az újságban ilyen megbízhatatlan emberkék. Egy kazetta manapság nem olcsó, főleg, ha programok is vannak rajta!

1989. augusztustól kéthavonta „Commodore Világ" címmel új folyóirat jelenik meg a „nagy játékosok"-nak. Ebben Amiga, C64 és C Plus/4-es játékleírásokat szándékoznak közölni. Ezért ezentúl feleslegesnek tartom a játékleírások közlését a Magazinban!

Lengyel István, Budapest

Örömmel vettem kezembe a Commodore Világ című kiadványt, és örömöm csak nőtt az indító cikk, a Bejelentkezés első sorait olvasva, mert egy Plus/4-et nyúzok magam is. Sajnos ez az öröm hamar elmúlt, sőt felháborodásba csapott át, mert a továbbiakban a cikk (?) „vérlázító" dolgokat tartalmazott. A szerzők szerint a Commodore-tulajdonosok igen sokan vannak (ez igaz is), „ . . . mégsincs nekik egy normáiis kiadványuk". Így, és ez a „sajtóhuba" is az idézet része. Ezek után felsorolják ezeket a „nem normális" kiadványokat: „Van ugyan egy Egyesületi Újság, meg Mikro Magazin — de erős a gyanúnk, hogy ezek nem igazán a közönség szája íze szerint készülnek."

Hát . . . Ez az egy mondat már több gondolatot is felvet az emberben. Ad 1. Az ún. „Egyesületi Újság" neve Commodore Újság, és mint ilyen, kifejezetten a Commodore-tulajdonosoknak szól. Ad 2. A „Mikro Magazin„ sem „Mikro Magazin", hanem Mikroszámítógép Magazin, esetleg Mikromagazin vagy Magazin. Ad 3. Ezek szerint a szintén ide tartozó Mikrovilág nem is létezik. Ad 4. A mondat második feléről és az újságírói etikáról már ne is beszéljünk . . .

A továbbiakban olvashatunk arról, hogy „Mi lesz ebben a kiadványban?" Lesznek például játékismertetők, C64-re és Amigára, mert ugye „ . . . Plus/4-re sajnos abszolút nem jelenik meg annyi játék (sőt, szinte semmi!), hogy nekik is tudnánk hasonló rovattal szolgálni". Ehhez képest a C16-os — Plus/4-es játék- és felhasználói programjait ismertető könyvekből már három jelent meg hazánkban, és a negyedik ilyen témájú könyv is hamarosan az üzletekbe kerül (talán már november végén).

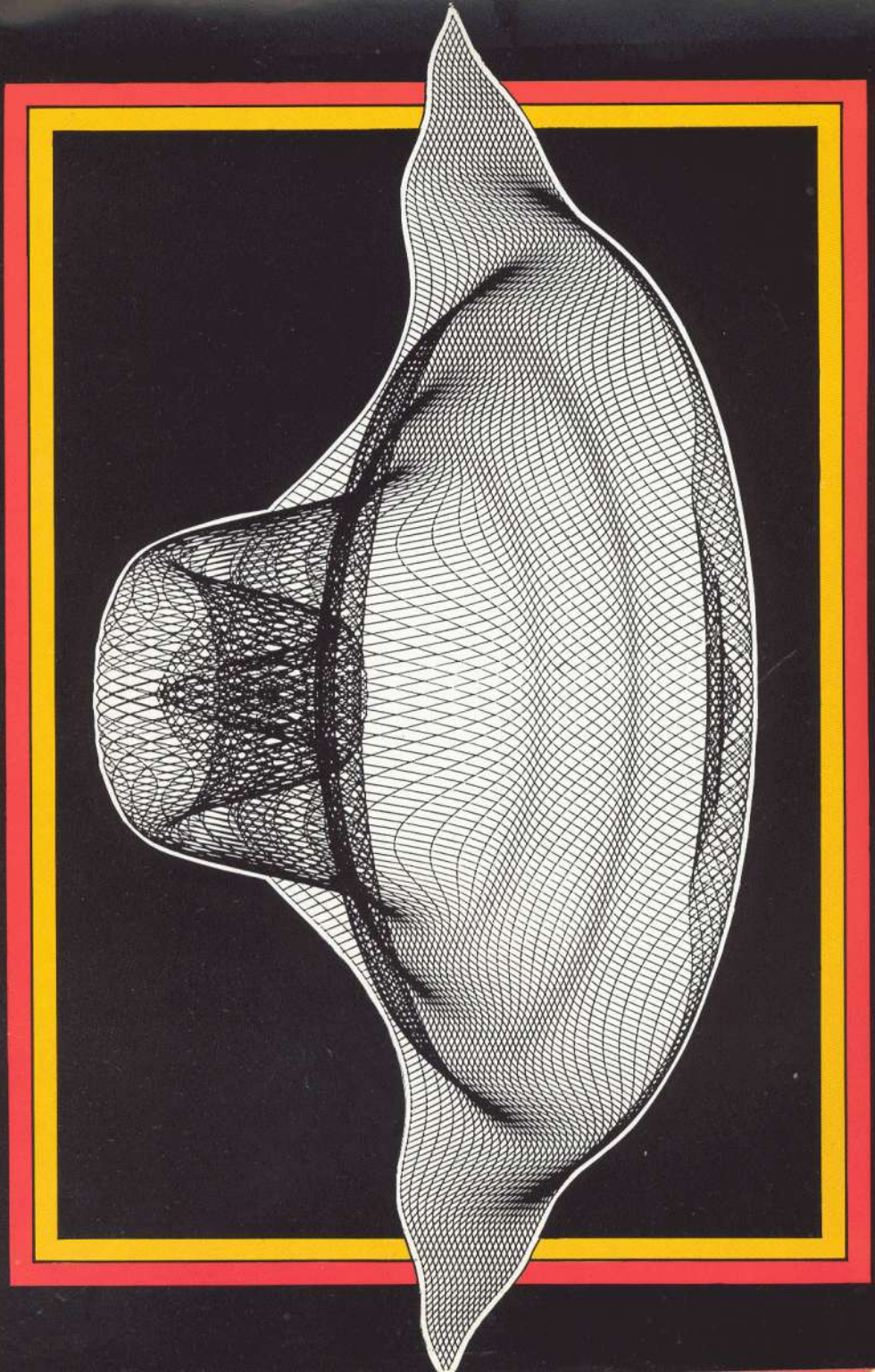
Ezek után következik a „Mi NEM lesz a kiadványban?" című bekezdés, amiben a számítógépes kiadványok végképp megkapják a magukét! Találva érezheti magát minden hasonló lap, és itt kiderül az is, hogy a „nem létező" Mikrovilág sem ismeretlen számukra. Például a lap szemére vetik a műholdas mellékletet „ . . . sőt a (ru)szkájcsennel műsorát sem fogjuk közölni . . . " — holott ezt a Mikrovilág terjedelemcsökkenés és árnövekedés nélkül közli! Tudtommal ilyen részletes műsor gyakorlatilag ingyen semmilyen más sajtóorgánumban nem található.

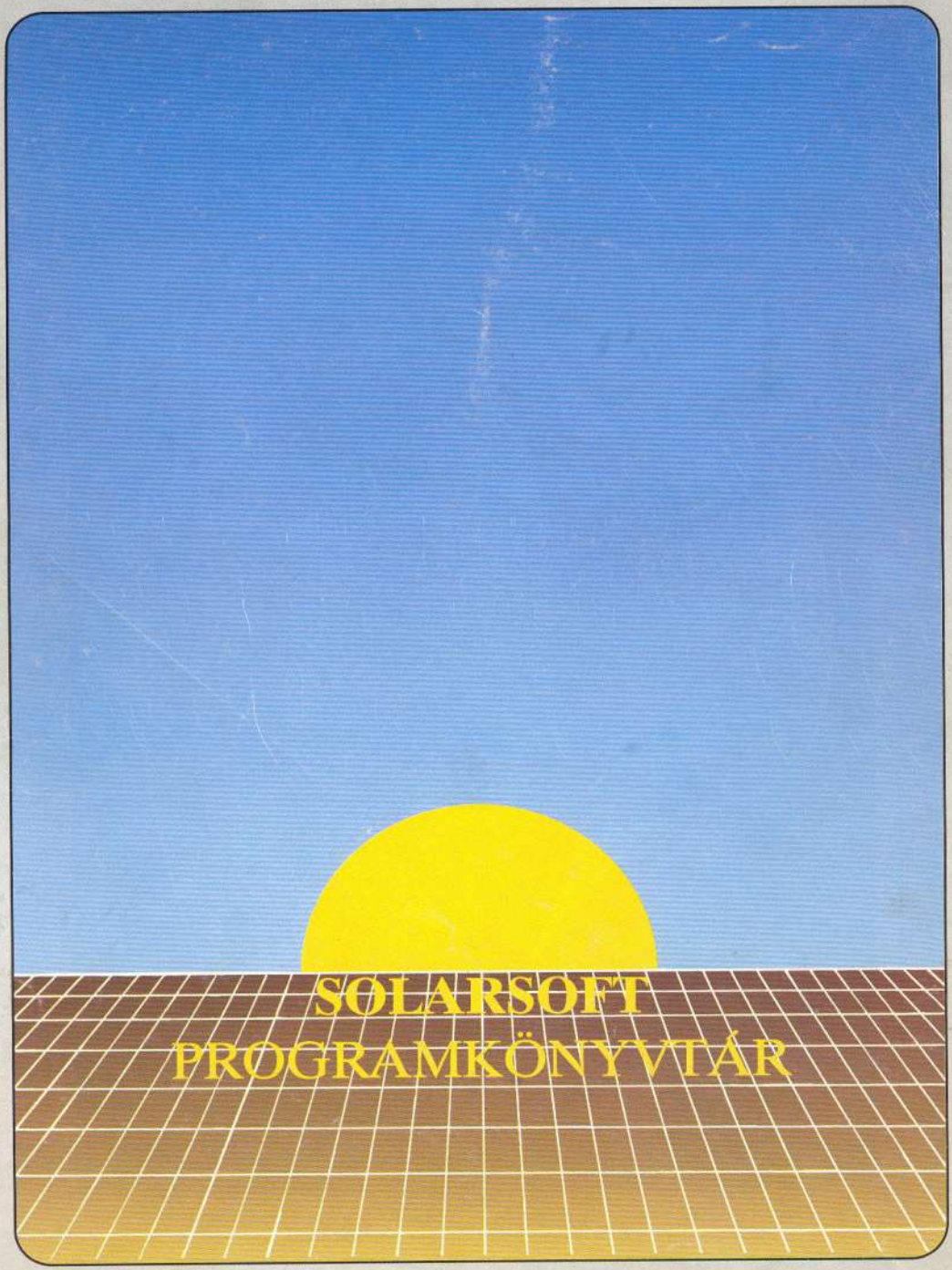
Végül a legfájóbb pont: az úgynevezett „Kazettaküldő Szolgálat". Ez valami nagyon „magyar" dolog. Egy postafiók címén lehet rendelni kazettákat és lemezeket, amelyeken Spectrum, Commodore és IBM PC programok vannak. Nem állíthatom biztosan, hogy ezek NEM jogtiszta programok, de erre lehet következtetni. Az egyik cikkben — The Last Ninja — például leírják az általuk is forgalmazott játék „letörésének" nehézségeit, ami miatt a tőlük származó kazettákról nehézkes a betöltés. Hát hurrá! Miközben a lapok — sőt ez a kiadvány is! — felhívják a figyelmet a vírusveszélyre, fő forrásként az illegális másolatokat és a „mókás kedvű" crackereket említve, itt egy ilyen akció, ami látszólag teljesen(?) legális! Kíváncsi lennék, a kiadványban felsorolt szoftverforgalmazók mit szólnának a Spectrum Világ — mert mint kiderült, rajtuk keresztül bonyolódik az „üzlet" — ajánlati listájához?

Először is elnézést, hogy a két levelet „összeházasítottam", de az érintett témák hasonlósága — úgy érzem — ezt indokolja. Sajnos az ifj. Fekete László által említett eset nem egyedi, sok ilyen és hasonló jelzést kaptunk. Amikor az Adok — veszek — cserélek — rovatot elindítottuk, célunk az volt, hogy olcsó hirdetési lehetőséget teremtsünk ahhoz — egy-egy hirdetés átlag 70—100 forintba kerül —, hogy egymástól elszigetelten működő körök, személyek megtalálják hasonló érdeklődésű partnereiket. Sajnos erre a csatornára ráleltek a szakma „hiénái" is; nem rendelkezünk nyomozó apparátussal, így nem tudjuk kiszűrni őket hirdetőink közül. Bízunk abban, hogy zömében a tisztességes partnerkeresők hirdetési fóruma leszünk.

A szerzői jogok lábbal tipróiról, a crackerek tevékenységéről is megvan a véleményünk, de — úgy tűnik — itt, ma Magyarországon a fellépés ellenük senkinek sem a dolga . . . A honi szoftveranomáliáknak természetes szülötte a cserebere. Ami a Commodore Világ c. lap megjelenését illeti, szerettük volna úgy üdvözölni, mint a palettát színesítő új laptársunkat. Erre a lap választott alapállását ismerve már kevesebb az esély. Nem tudom, a Mikrovilág mit szól hozzá, de ránk nézve sem éppen hízelgő ez az „entrée". Kutya-kötelességem vitába szállni azzal a kijelentéssel, hogy „ezek nem igazán a közönség szája íze szerint készülnek". Hát akkor miért olvassa 17 000 ember még mindig a Mikromagazint? Ebben a szakmában csak az „ismeretlen" Mikrovilág rendelkezik hasonló példányszámmal. Valami — a közönség szája íze szerint való — csak lehet bennük, nemde?

Leveleiket, észrevételeiket továbbra is várjuk, hogy profitálhassunk belőlük a lap összeállításánál.





A katalógus első része a 42. oldalon olvasható.